

21世纪高等院校通识教育规划教材

LaTeX KEJI LUNWEN XIEZUO JIANMING JIAOCHENG

LaTeX科技论文写作简明教程

王伊蕾　李涛　编著
Wang Yilei　Li Tao



清华大学出版社

21世纪高等院校通识教育规划教材

LaTeX 科技论文写作简明教程

王伊蕾　李　涛　编著

清 华 大 学 出 版 社

北　京

内 容 简 介

本书全面介绍 LaTeX 的安装和使用过程，对于论文题目、作者信息、摘要等基本信息做了简单介绍，内容包括图形的插入、表格的插入、数学公式的输入和参考文献等方面，详细论述了各个方面可能碰到的问题并给出相应的解决方案，最后给出一些常见的错误，以便读者查阅。本书通过大量实例，向读者展示了如何使用 LaTeX 命令，使读者有一个直观、清晰的认识。

本书简明介绍科技文献中图片、表格、数学公式和参考文献的常见的问题，为科技工作者撰写论文提供简洁的工具，提供了多种期刊模板实例，结合作者多年排版经验，为科技工作者提供任务驱动型的 LaTeX 简明教程。本书适合作为高等院校计算机、数学专业高年级本科生、研究生的教材。

图书在版编目（CIP）数据

LaTeX 科技论文写作简明教程/王伊蕾，李涛编著. --北京：清华大学出版社，2015
21 世纪高等院校通识教育规划教材
ISBN 978-7-302-42169-6

Ⅰ. ①L… Ⅱ. ①王… ②李… Ⅲ. ①排版－应用软件－应用－科学技术－论文－写作－高等学校－教材 Ⅳ. ①H152.3-39

中国版本图书馆 CIP 数据核字(2015)第 271820 号

责任编辑：白立军
封面设计：傅瑞学
责任校对：焦丽丽
责任印制：杨　艳

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：北京国马印刷厂
经　　销：全国新华书店
开　　本：185mm×230mm　　**印　　张**：7.5　　**字　　数**：142 千字
版　　次：2015 年 11 月第 1 版　　**印　　次**：2015 年 11 月第1 次印刷
印　　数：1～2000
定　　价：25.00 元

产品编号：065645-01

前　言

LaTeX是目前流行和使用最为广泛的TeX宏集，它构筑在Plain TeX的基础之上，并融入了很多功能，使用者可以方便地利用TeX的强大功能。对于生成复杂的数学公式，LaTeX的表现更为出色。TeX是LaTeX的基础，它是产生于20世纪70年代末到20世纪80年代初的一款计算机排版软件，用来排版高质量的书籍，特别是包含有数学公式的书籍。TeX的初衷是用于个人的一个排版软件，基于这一点，TeX主要是由书籍或论文的作者本人来使用。TeX有许多方便作者的自定义功能，使用也简单方便，即使作者不具备太多的编程知识，也可以自己排版，因此TeX受到用户的青睐。

LaTeX的优势在于，在给定模板的情况下，只需要输入内容即可，不用考虑排版和页面设置等问题。另外对于毕业论文来说，参考文献过多并且在编辑的过程中，经常会有章节变动和参考文献编号变动的问题。不断地调整编号是件烦琐的事情，但是LaTeX可以很方便地处理自动编号的问题。另外，对于数学公式的排版，LaTeX也具有很大的优势，排版出来的数学公式非常规范、美观。

本书的目标对象主要是：需要经常使用LaTeX软件排版论文的高校师生和科研院所的科研人员。本书以LNCS模板为例，以撰写一篇学术论文为主要描述路线，从模板的选择、题目、摘要、正文、图形、表格、数学公式、参考文献等几个方面说明了如何使用LaTeX有效地生成一篇科技论文。

本书是一本简明教程，LaTeX的功能强大，显然不局限于本书所列的几项功能。LaTeX除了专业科技论文，最终生成PDF格式的文件以外，还可以实现幻灯片制作等其他功能，在这里不一一赘述。本书假定读者是LaTeX的初学者，希望本书能够达到以下几个目标。

(1) 了解LaTeX的基本架构，学会LaTeX的安装和基本的菜单使用技巧。

(2) 了解各种模板的使用，了解宏包的功能，熟悉LaTeX的页面设置命令。

(3) 掌握各种图形的插入命令,熟悉图形属性的设置以及子图的设置等命令。

(4) 掌握表格的插入命令,表格字体的设置命令、表格自动编号以及表格引用。

(5) 掌握数学公式的输入命令,了解数学公式命令以及各种宏包的对应关系。

(6) 掌握参考文献的引用和设置命令。

(7) 了解各种错误的原因以及解决方案。

本书仅涵盖了 LaTeX 很小一部分内容,适合于编辑简单的 LaTeX 文档。然而 LaTeX 是一个开放的系统,与其他软件一样,它是在不断进化的。不仅其内核从最初的 LaTeX 2.09 到 LaTeX 2e 再到正在开发中的 LaTeX 3,而且还有数以千计的工具宏包在不断更新,完成各种复杂的排版功能。

为了增强本书的可读性,笔者给出大量实例,这些实例都是笔者对多年来撰写科研论文过程中遇到的实际问题的总结,本书包含了笔者多年来使用 LaTeX 的一些心得。本书还总结了在撰写论文时经常遇到的一些问题,这些问题也是科技工作者在撰写论文过程中或多或少会遇到的。希望本书能够给初学者一些帮助,对于一些简单的问题,能够给出答案。

全书共分 7 章:第 1 章介绍 LaTeX 的产生背景,LaTeX 的各个版本,以及它们之间的区别;第 2 章介绍 LaTeX 的模板,其中包括 LaTeX 的基本框架、常用宏包以及页面设置;第 3 章介绍 LaTeX 中图形的插入,首先介绍了图形的类型,本书着重介绍了 EPS 和 PDF 格式图形的插入命令,然后介绍了图形属性的设置,主要是图形位置和图形大小的设置,接着介绍了子图插入的问题,最后简单介绍了图形通栏的设置情况;第 4 章介绍表格的基本命令,表格中字体大小的设置以及表格在正文中引用的方式;第 5 章介绍 LaTeX 中数学公式和数学符号的使用问题,首先介绍了数学符号的使用,然后介绍了数学公式显示的问题,接着针对长数学公式分行显示和编号的问题进行介绍,另外本章还对数学公式中字体大小、数学环境的定义进行了描述,最后给出了一些常见的数学符号命令;第 6 章介绍参考文献格式的设置问题,给出了两种常用的参考文献设置方式;第 7 章给出一些常见的错误类型和解决方案。

本书由李涛编写了第 1 章～第 3 章并负责校对全书的初稿,王伊蕾编写了第 4 章～第 7 章并负责本书结构框架的制定。本书的编写还得到鲁东大学邹海林教授、山东大学

计算机科学与技术学院徐秋亮教授的精心指导和热情帮助,在此表示衷心的感谢。另外,秦海荣和陈鲁丰在本书早期的资料收集和整理方面给予了很多无私帮助,在此一并表示感谢。本书的出版还得到山东省优秀中青年科学家科研奖励基金(No. BS2014DX016)和国家自然基金(No. 61502218)的资助。

清华大学出版社对于本书的立项与出版给予了极大帮助,对于责任编辑及其他参与此书编辑工作的各位老师为本书顺利出版而付出的辛勤劳动表示由衷的感谢。

由于作者水平有限,书中不足之处在所难免,恳请广大读者和同行批评指正。

王伊蕾

2015 年 6 月

目　录

第1章　引言

自从 LaTeX 问世以来，流传最广的版本是 LaTeX 2.09。由于 LaTeX 的众多优点，在计算机科学、数学及相关学科得到广泛的应用，吸引了许多专家、爱好者为其编写及添加了各式各样的宏包和宏库，例如 PostScript 字体处理、排版复杂数学公式的 AMSLaTeX 等，这使得 LaTeX 的功能不断地扩充，应用领域不断地扩大。但是，由于没有统一的宏包编写规划和编写格式，造成某些宏包的功能彼此接近，而命令相互冲突。这导致同一个源文件在某种格式的 LaTeX 中能够完美运行，而在另一种格式中就可能编译出错或结果有所不同。很多网站和编辑部为了处理不同来源的 LaTeX 文件，不得不置备各种格式的 LaTeX 系统。有些宏包很难分辨出是为哪种格式编写的，需要反复尝试。

1.1　LaTeX 产生背景

LaTeX(音译“拉泰赫”)是由美国计算机学家莱斯利·兰伯特(Leslie Lamport)在 20 世纪 80 年代初期开发的一种排版系统。它是当今世界上最流行和使用最为广泛的 TeX 宏集。LaTeX 在 Plain TeX 的基础上增加了很多功能，使用者可以更加方便地运用 TeX 的强大功能完成一些文档编辑任务。使用 LaTeX 的优势在于，用户无须自己设计命令和宏等，即使用户没有排版和程序设计的背景知识，也可以熟练应用 TeX 提供的强大功能。因此，即使用户并不是很了解 TeX，也可以在几天，甚至几小时内生成很多具有书籍质量的印刷品。对于复杂的表格和数学公式，LaTeX 的表现更为出色，因此它非常适用于生成高印刷质量的科技和数学类文档。这个系统同样适用于生成从简单的信件到完整书籍的所有其他种类的文档。LaTeX 使用 TeX 作为它的格式化引擎，当前的版本是 LaTeX2e。

LaTeX 自从 20 世纪 80 年代初问世以来，也在不断地发展。最初的正式版本为 LaTex 2.09，经过几年的发展，许多新的功能、机制被引入到 LaTeX 中。在享受这些新功能带来

便利的同时，它所伴随的副作用也开始显现，这就是不兼容性。标准的 LaTeX 2.09 引入了“新字体选择框架”(NFSS)的 LaTeX、SLiTEX 和 AMS-LaTeX 等，它们相互之间并不兼容。这给使用者和维护者都带来很大的麻烦。为结束这种糟糕的状况，Frank Mittelbach 等人成立了 ATeX3 项目小组，目标是建立一个最优的、有效的、统一的和标准的命令集合。这是一个长期目标，向这个目标迈出的第一步就是在 1994 年发布的 LaTeX2e。

TeX 还只是着重在于如何排版的层次上，而不是从一位作者的立场出发。对它深层功能的进一步发掘，需要相当丰富的编程技巧。因此它的应用就局限于高级排版和程序设计人员。虽然 TeX 的功能强大，用它可以排版任何样式的文稿，但灵活掌握 TeX 的 900 条初始命令，对于没有任何编程基础的普通作者还是有困难的。因此，在 TeX 公开几年后，利用 TeX 的宏定义功能开发的宏库 AMS-LaTeX 和 LaTeX 应运而生。

AMS-LaTeX 是受美国数学学会委托编写的，主要用于美国数学学会及其分支机构出版的大量书籍、期刊和评论。AMS-LaTeX 目前的版本为 2.0，AMS-LaTeX 中含有一个宏包供作者使用。用 AMS-LaTeX 可以方便地排印非常复杂的数学公式和 AMS 制定的全部数学符号。AMS-LaTeX 包括两部分：一部分是 amsmath 宏包，主要用来排版数学符号和公式；另一部分是 amscls 宏包，提供了美国数学学会要求的论文和书籍的格式。在提供了 AMS-LaTeX 的同时，美国数学学会还提供一套数学符号的字库：AMSFonts。这套字库中增加了很多 TeX 的标准字库 Computer Modern 所没有的一些数学符号，如粗体数学符号等。AMSFonts 现在的版本为 2.2，提供 Metafont 和 Type1 两种字库下载。

尽管在排版数学公式和数学符号方面，LaTeX 不如 AMS-LaTeX，但是 LaTeX 提供了大量易于学习和使用的命令，例如，非常有用的交叉引用命令等，都是 AMS-LaTeX 所不具备的。因而，LaTeX 的用途更为广泛，特别是在排版信件、书刊、诗集等方面更优于 AMS-LaTeX。

LaTeX 是 TeX 中的一种格式(format)，是建立在 TeX 基础上的宏语言，也就是说，每一个 LaTeX 命令实际上最后都会被转换解释成几个甚至上百个 TeX 命令。但是，普通用户可以无须知道这中间的复杂联系。正如编程时使用一些已经编译好的函数库和模板可以使用户仅用几条命令就实现很多复杂功能一样，LaTeX 根据人们排版文章的习

惯,定义了很多命令和模板,通过这些命令和模板,用户可以很快得到漂亮的排版结果。图 1.1 表示了 LaTeX 和 TeX 之间的联系。

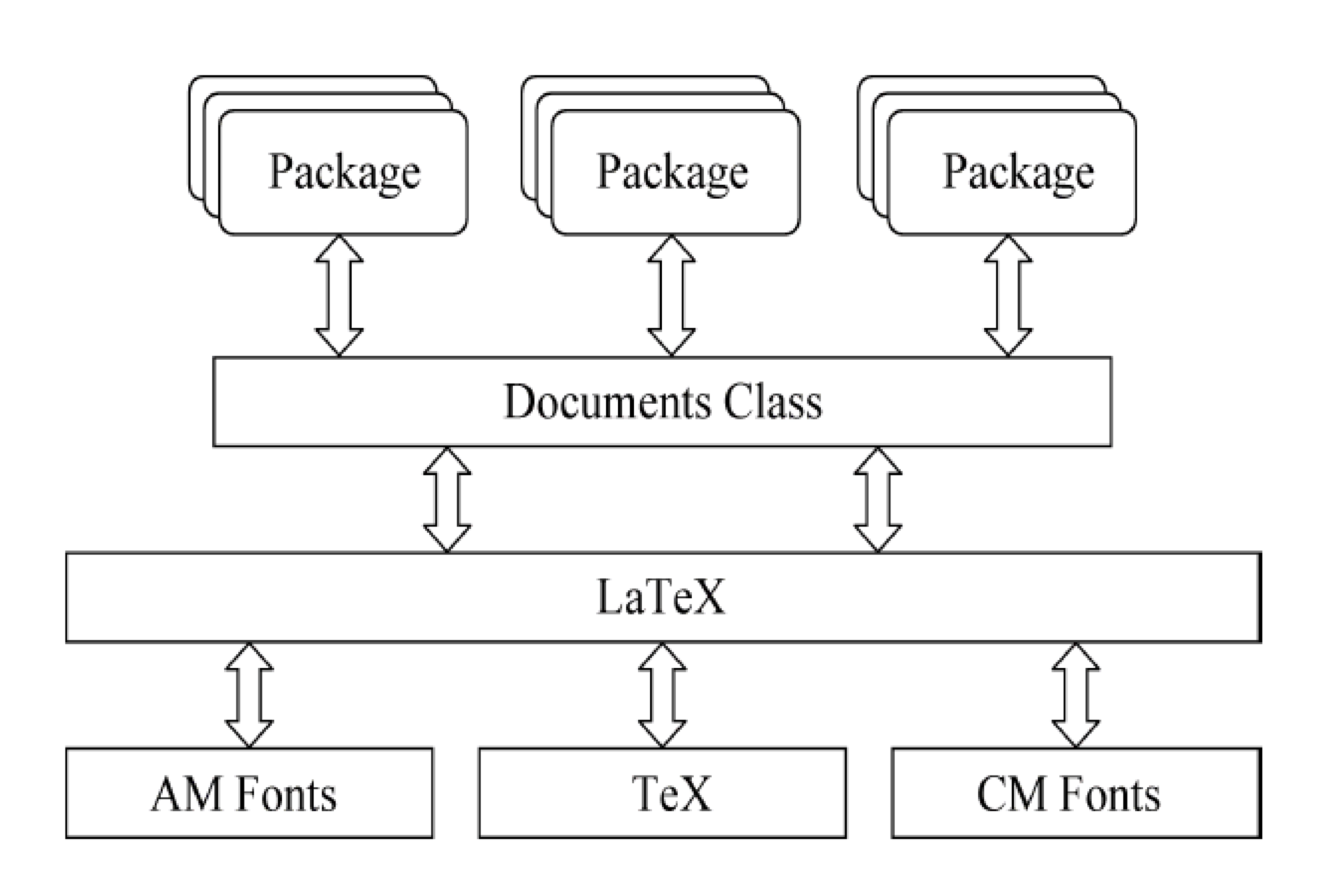


图 1.1　LaTeX 和 TeX 之间的联系

1.2　LaTeX 版本介绍

1.2.1　TeX

最基本的 TeX 程序只是由一些很原始的命令组成,它们可以完成简单的排版操作和程序设计功能。然而,TeX 也允许用这些原始命令定义一些更复杂的高级命令。这样就可以利用低级的块结构,形成一个用户界面友好的环境。在处理器运行期间,该程序首先读取格式文件,其中包含各种以原始语言写成的高级命令,也包含分割单词的连字号安排模式。接着处理程序就处理源文件,其中包含要处理的真正文本,以及在格式文件中已定义了的格式命令。创建新格式是一件需要由具有丰富知识的程序员来做的事情。把定义写到一个源文件中,这个文件接着被一个称为 iniTeX 的特殊版本的 TeX 程序处理。它采用一种紧凑的方式存储这些新格式,这样就可以被 TeX 程序很快地读取。

1.2.2　PlainTeX

Knuth 设计了一个名叫 PlainTeX 的基本格式,以与低层次的原始 TeX 呼应。用 TeX 处理文本时,这种格式是基本部分之一,以至于用户有时分不清到底哪条指令是真正

的处理程序 TeX 的原始命令,哪条是 PlainTeX 格式的。大多数声称只使用 TeX 的人,实际上指的是只用 PlainTeX。PlainTeX 也是其他格式的基础,这进一步加深了很多人认为 TeX 和 PlainTeX 是同一事物的印象。PlainTeX 的重点还只是在于如何排版的层次上,而不是从一位作者的观点出发。对它的深层功能的进一步发掘,需要相当丰富的编程技巧。因此它的应用就局限于高级排版和程序设计人员。

1.2.3 LaTeX2e

Frank Mittelbach 等人成立了 LaTeX3 项目小组,目标是建立一个最优的、有效的、统一的、标准的命令集合,即得到 LaTeX 的一个新版本。1994 年发布的 LaTeX2e 采用了 NFSS 作为标准,加入了很多新的功能。LaTeX2e 每 6 个月更新一次,修正发现的错误并加入一些新的功能。在 LaTeX3 最终完成之前,LaTeX2e 将是标准的 LaTeX 版本。

LaTeX2e 对 LaTeX 2.09 做了很大改进,增加很多新功能。从文件内容上看,两者最显著的不同在于 LaTeX 2.09 使用\documentstyle 命令定义文档类型以及所包含宏包,如\documentstyle[twoside,epsfig]{article},而 LaTeX2e 使用\documentclass 命令定义文档类型,用\usepackage 命令包含宏包,例如:

```
\documentclass[twoside]{article}
\usepackage{epsfig}
```

LaTeX 2.09 文件一般都可以在 LaTeX2e 系统中以兼容方式编译。但是兼容方式编译速度慢,而且很多 LaTeX2e 的新功能无法使用。如果不需要编译以前的 LaTeX 2.09 文件,无须使用 LaTeX 2.09,也不用知道 LaTeX 2.09 与 LaTeX2e 的差别。大部分 LaTeX 2.09 文件都可以通过用\documentclass 命令和\usepackage 命令代替\documentstyle 命令修改为 LaTeX2e 格式。有时可能需要一些特殊宏包,例如 latexsym,对旧的 LaTeX 2.09 命令提供支持。

TeX 在不同的硬件和操作系统上有不同的实现版本。类似于 C 语言,在不同的操作系统中有不同的编译系统,例如在 Linux 操作系统下使用 gcc,在 Windows 操作系统下使用 Visual C++ 等。有时,也允许在一种操作系统中存在多种 TeX 系统。目前常见的 UNIX/Linux 下的 TeX 系统是 Texlive,Windows 下则有 MiKTeX 和 fpTeX。CTeX 指

的是 CTeX 中文套装的简称，是把 MiKTeX 和一些常用的相关工具，如 GSview、WinEdt 等包装在一起制作的一个简易安装程序，并对其中的中文支持部分进行配置，使得安装后马上就可以使用中文。

1.3　LaTeX 的安装

CTeX 的安装文件 CTeX_2.9.0.152_Full.exe 可以从通过网址 http://jcube.sjtu.edu.cn/f/588Z6u14k06i5PS8 获得。下载安装文件后，双击 exe 文件即可安装，如图 1.2 所示。

可以选择安装语言，可选的语言包括 Chinese(Simplified)和 English，如图 1.3 所示。

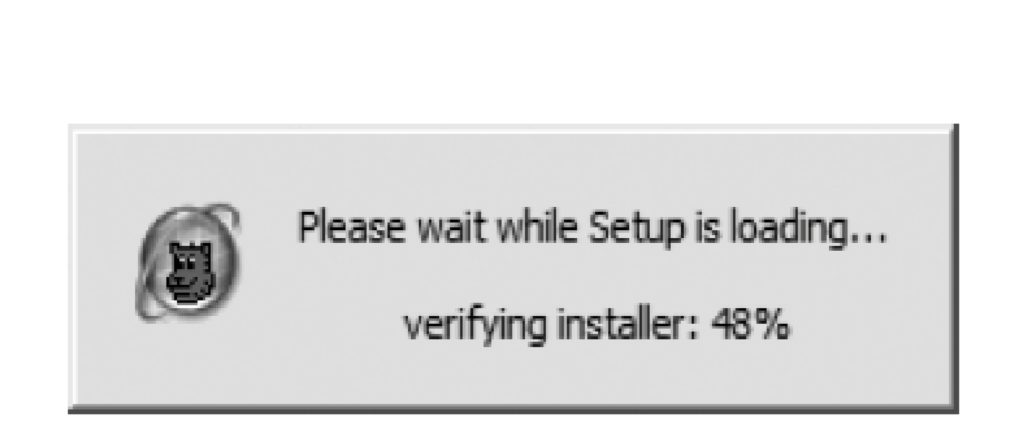


图 1.2　安装等待进度条



图 1.3　选择安装语言

选择好语言后单击 OK 按钮，进入 CTeX 2.9.0 的安装向导界面，如图 1.4 所示。

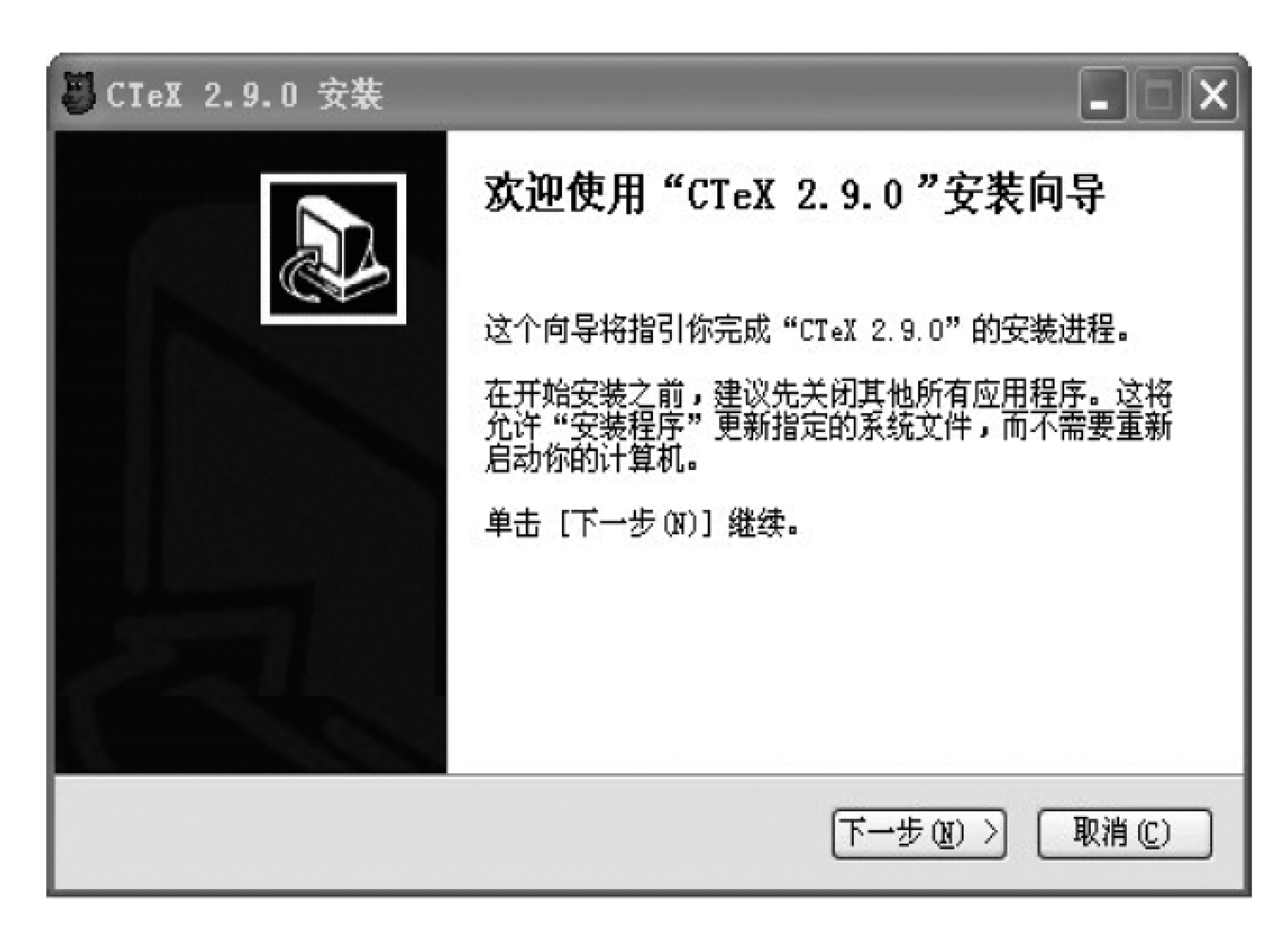


图 1.4　CTeX 2.9.0 安装向导

单击“下一步”，进入图 1.5 所示的“授权协议”部分。在这一部分，需要作者仔细阅读“授权协议”，如果不同意协议内容，可以选择“取消”。但是注意：在这一步，必须接受协议，并且单击“我接受”按钮，才可以进入下一步。

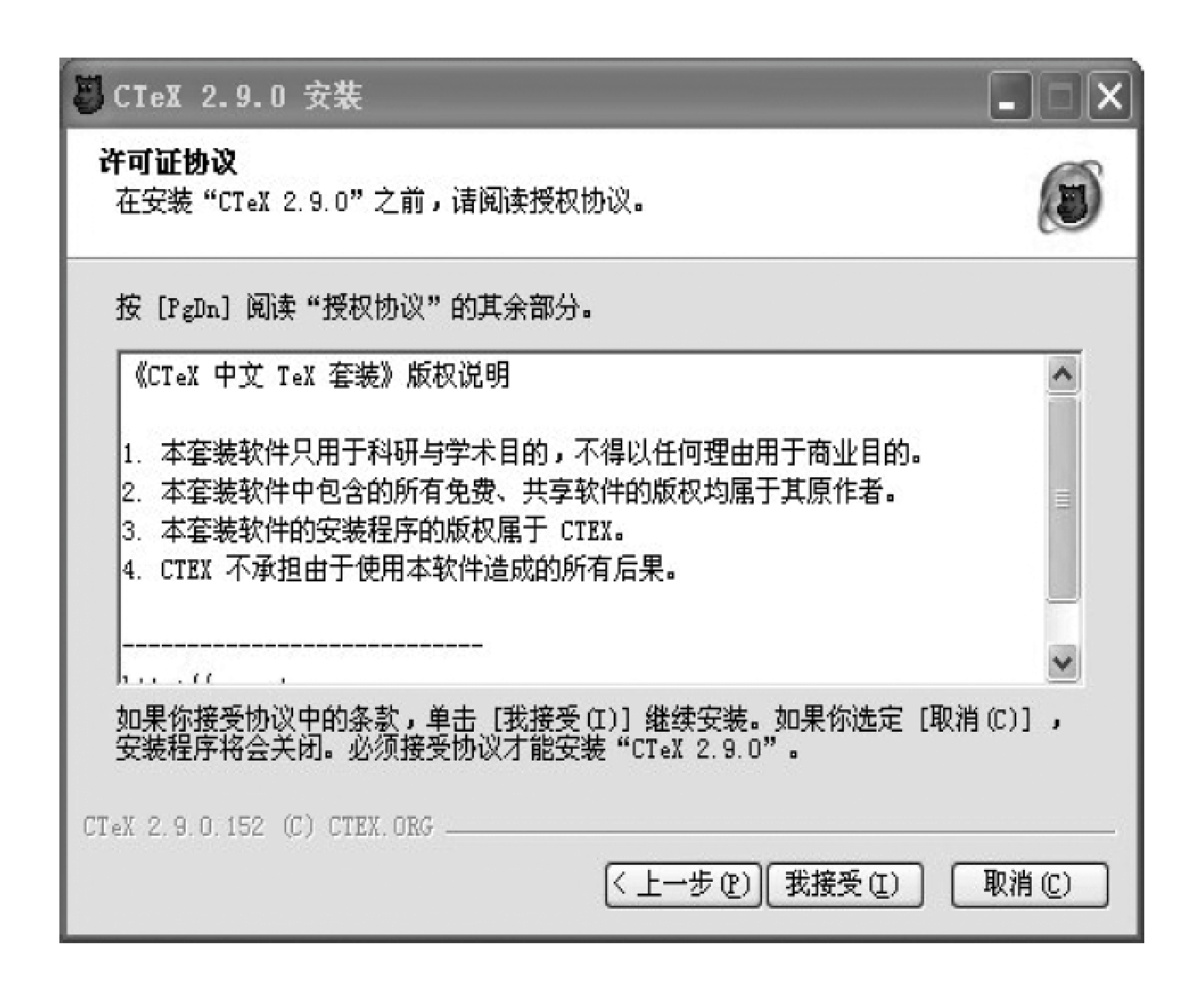


图 1.5 CTeX 2.9.0 安装协议

单击“我接受”按钮后，即可进入如图 1.6 所示的界面。在图 1.6 中根据用户不同的需求，可以选择不同的组件。

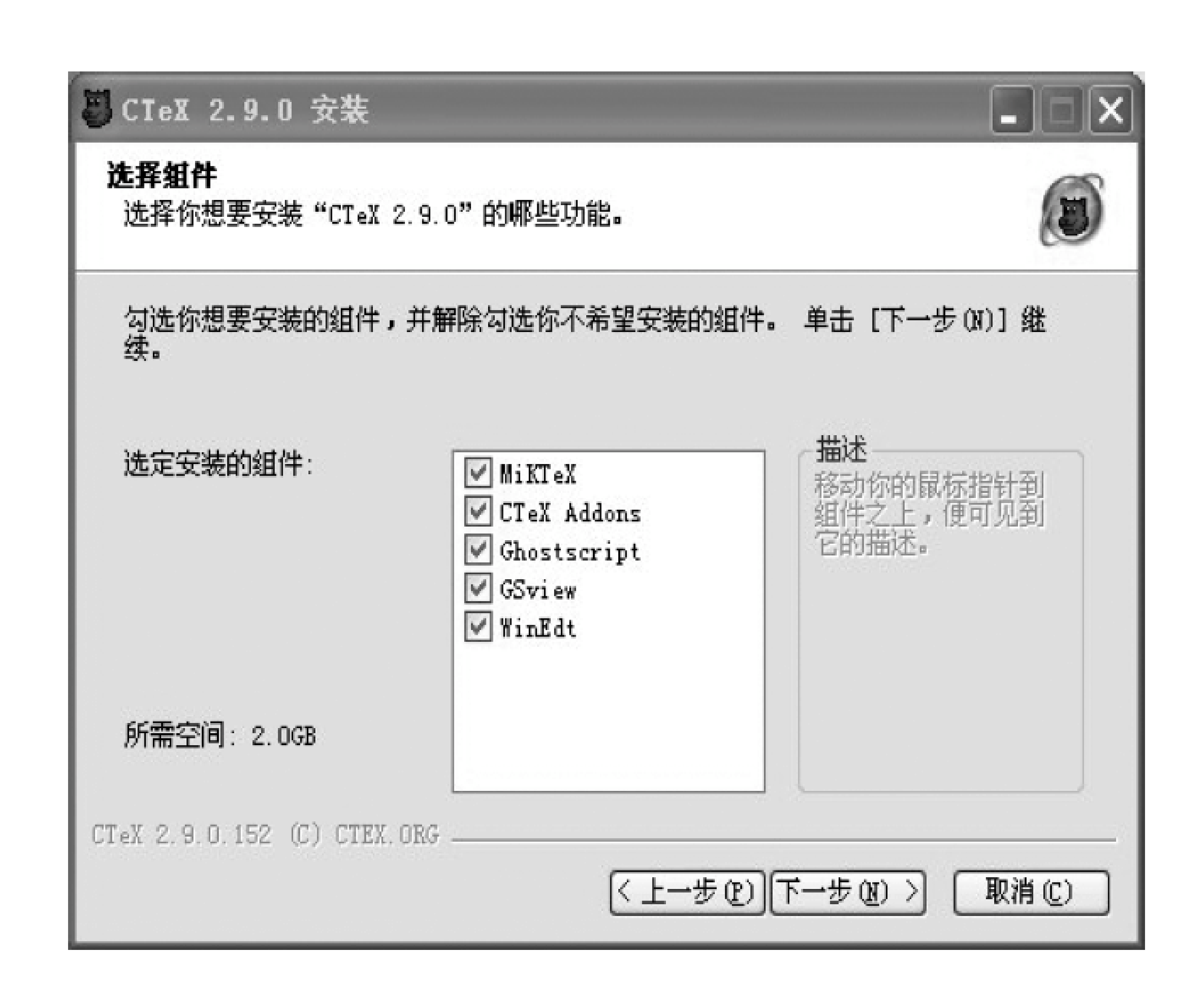


图 1.6 组件选择

(1) MiKTeX 组件。Windows 下最好用的 TeX 系统之一,带有一个很优秀的 DVI 预览器 Yap。

(2) CTeX Addons 组件。中文 TeX 组件,包括 CJK/CCT/TY 和相应的字体设置,以及一些中文 LaTeX 宏包。

(3) Ghostscript 组件。PS(PostScript) 语言和 PDF 文件解释器,可在非 PS 打印机上打印它们。可以将 PS 文件和 PDF 文件相互转换。

(4) GSview 组件。GSview 是 Ghostscript 的图形界面程序,通过 Ghostscript 的支持,可以方便地浏览和支持 PS 文件。

(5) WinEdt 组件。WinEdt 是一个编辑器,它内置了对 TeX 的良好支持。在它的菜单和按钮上,可以直接调用 TeX 程序,包括编译和预览等。WinEdt 还能帮助用户迅速输入各种 TeX 命令和符号,用户不必记忆复杂命令。

建议初次使用 LaTeX 的用户将这些选项都选上。单击"下一步"按钮,选择安装目录,如图 1.7 所示。

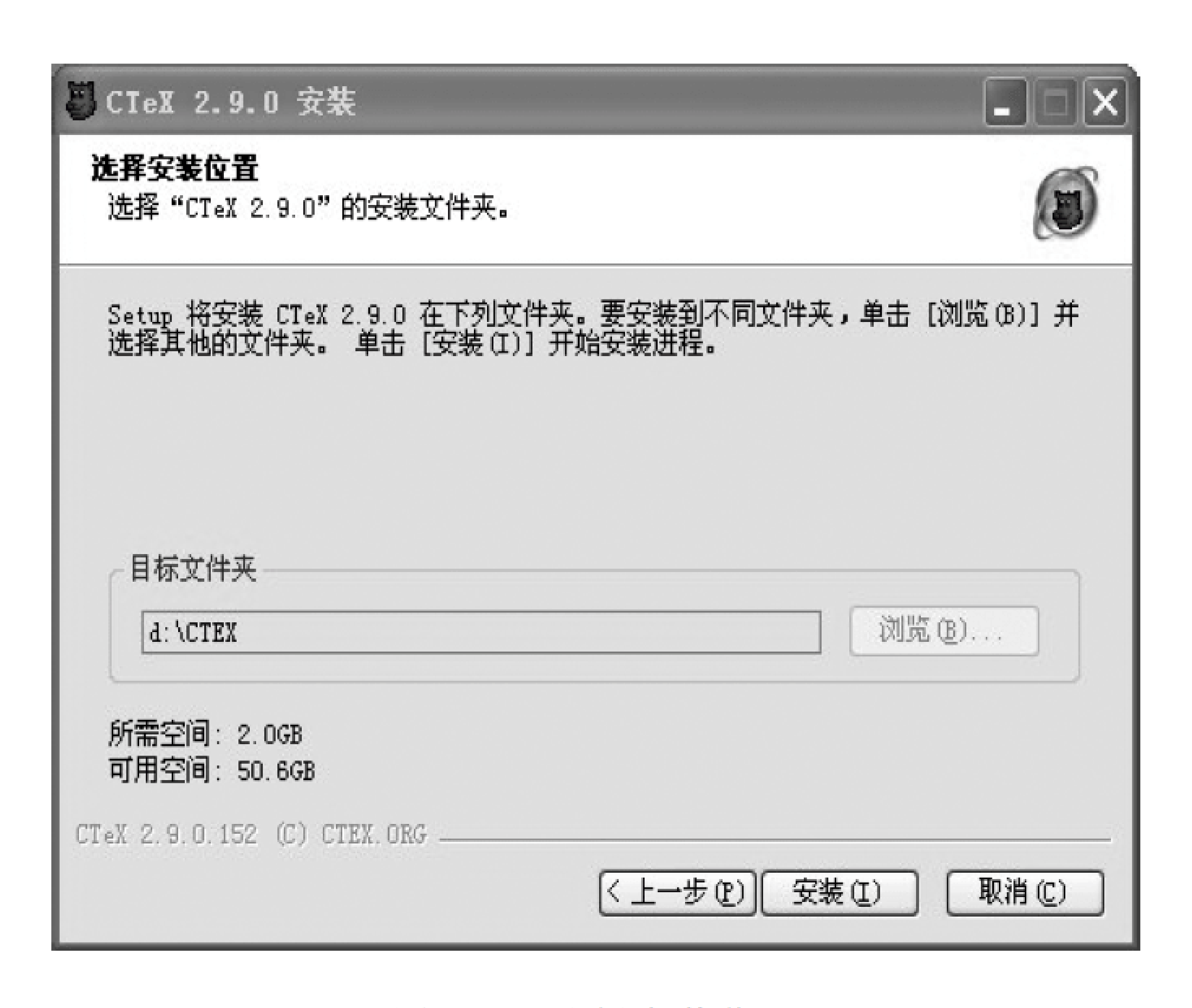


图 1.7　选择安装位置

如果不选择默认位置,可以单击"浏览"按钮选择合适的安装目录。确定好安装目录后,单击"安装"按钮,进入安装文件界面,如图 1.8 所示。



安装进度界面(一)



安装进度界面(二)

图 1.8　安装进度

安装结束后，单击“下一步”按钮，进入安装完成界面，如图 1.9 所示。

安装成功之后，就可以使用 LaTeX 编辑文件了。LaTeX 的操作主界面如图 1.10 所示。LaTeX 的操作界面和一般的 Windows 窗口类似，也包括菜单栏和工具栏，另外还有

一些快捷菜单，这里就不详细描述了。



图 1.9　安装完成界面

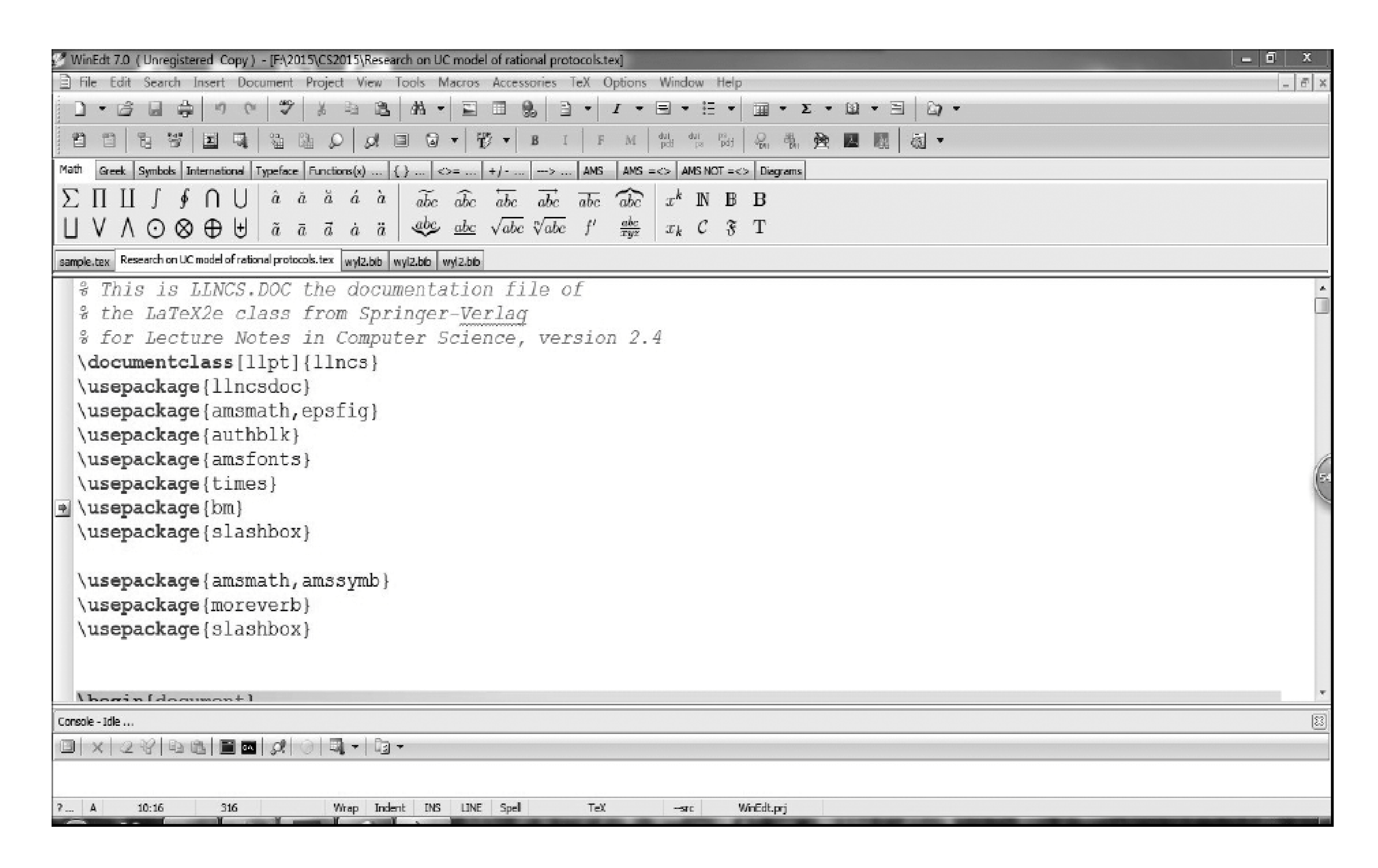


图 1.10　LaTeX 操作主界面

使用 LaTeX 可以编译生成 DIV、PDF 和 PostScript 类型的文件，它们对应的编译工具如下。

（1）tex：编译 TeX 的源文件生成 DVI 文件。

（2）pdftex：编译 TeX 的源文件生成 PDF 文件。

（3）latex：编译 LaTeX 的源文件生成 DVI 文件(经常使用的一个命令)。

（4）pdflatex：编译 LaTeX 的源文件生成 PDF 文件。

（5）dvi2ps：将 DVI 文件转换成 PostScript 文件。

（6）dvipdf：将 DVI 文件转换成 PDF 文件。

当一个 LaTeX 文件建立之后，编译该文件只能再生成 DVI 格式的文件，然后由 pdftex 生成 PDF 文件。

注意：PDF 文件可以由 pdftex 生成也可以由 dvipdfm 生成，目前一般习惯使用 pdftex 生成 PDF 文件。Dvipdfm 是由 DVI 文件转化成 PDF 文件，而 DVI 格式是一种比较老的文件格式，它不支持超链接。

图 1.11 显示了 TeX 和 LaTeX 源文件以及其他格式文件之间的关系。

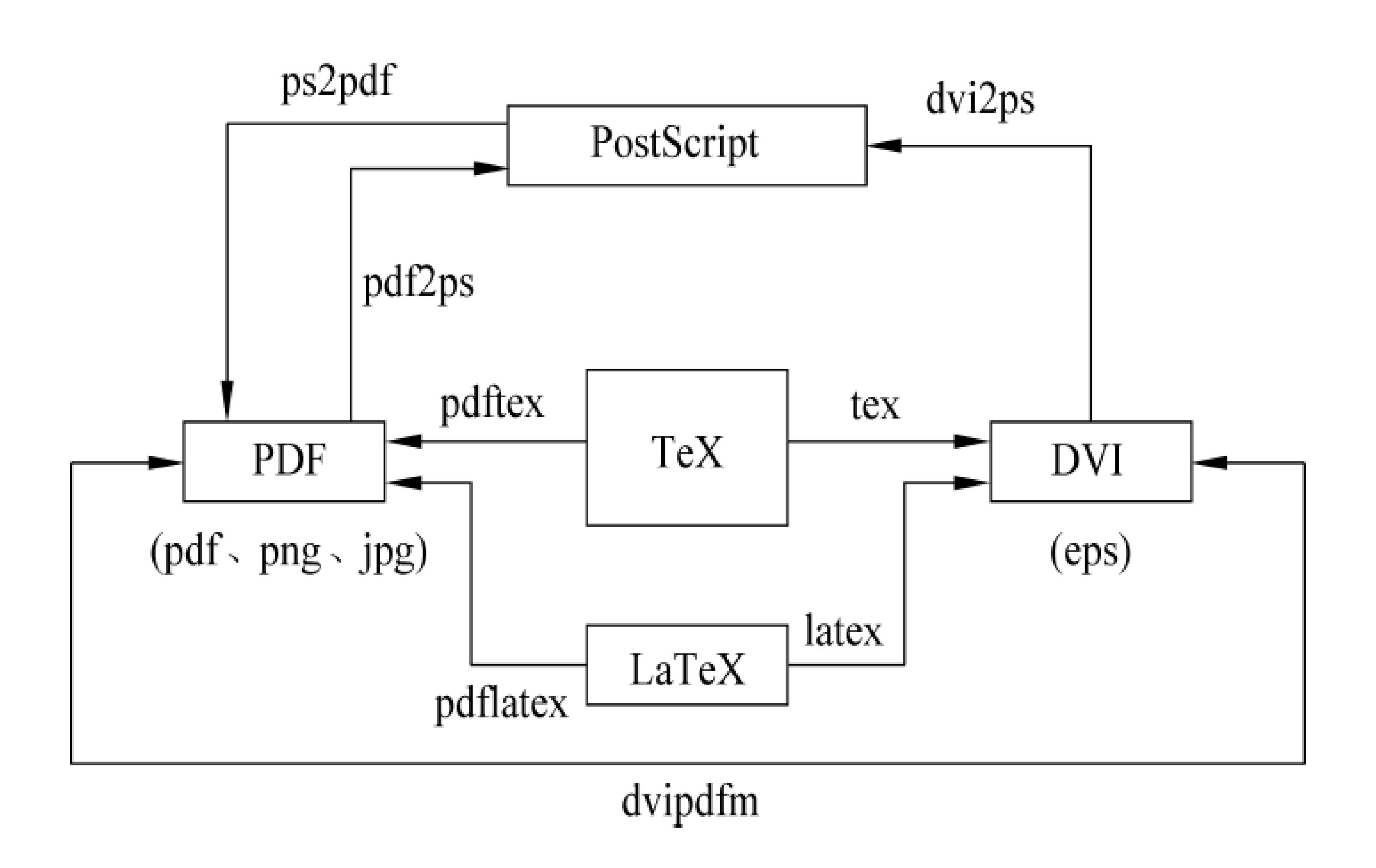


图 1.11　源文件以及各类型文件之间的关系

第 2 章　LaTeX 模板介绍

2.1　LaTeX 基本的框架

LaTeX 基本的框架如图 2.1 所示。

(1) \documentclass[选项]{类},确定整篇文章的处理格式,期刊或者会议论文一般可选类为 article,再附上控制全局格式的选项,比如字体、字号、页面格式、纸张大小等。也有期刊直接提供类模板,比如 *Lecture Notes in Computer Science* 提供的模板为 lncs.cls,只要把相应的类名 lncs 放到{类}里就可以了,不需要自己去费神。需要注意的是,lncs.cls 这个文件需要放在当前文件夹下,否则会因为找不到该文件提示出错。如果使用 article 类型,则不需要有对应的 cls 格式的文件,因为是系统默认的类型。

```
\documentclass{lncs}
\begin{document}
This is the body of the article.
\end{document}
```

```
\documentclass{article}
\begin{document}
\title{…}
\author{…}
\institute{…}
\maketitle
\begin{abstract}
  ⋮
\end{abstract}
\keywords{…;…;…}
\section{…}
\subsection{…}
\subsection{…}
\section*{Acknowledgements}
\begin{thebibliography}
\bibitem{}
\bibitem{}
  ⋮
\end{thebibliography}
\end{document}
```

图 2.1　LaTeX 基本的框架

(2) 接下来是包含一些使用的宏包来增强功能,\usepackage{宏包},宏包包含在.sty 文件中,主要的宏包如下。

① CJK 支持中文环境,对应宏包命令为\usepackage{CJK}。如果想在 LaTeX 中使用中文,需要使用这个宏包。

② 如果在文章中使用 Times New Roman 字体，需要用到宏包命令\usepackage{times}。

③ 如果在文章中需要插入图片，要用到的宏包是 graphicx，对应的宏包命令为\usepackage{graphicx}。

④ 如果希望制作成的 PDF 格式文件中具有超文本链接功能，例如目录、参考文献、图形和表格等，需要在 LaTeX 源文件相应的位置中有交叉引用的地方。此时需要使用宏包命令为\usepackage{pyperref}完成引用超链接。

也有期刊提供宏包来定制格式，比如 *IEEE Computer Society Press*。其功能类似于 C 语言中的♯include，可以为第三方提供接口。有些提供的样例文件中在 documentclass 的选项中添加宏包，这是与老版本兼容。

(3) 以上为导言区，接着余下的都是正文部分，包含在\begin{document}和\end{document}内。LaTeX 命令的作用对象和范围与 HTML 的标签有点类似，有开始和结束标志，开始位置内会定义一些表现格式。导言区还可能有\pagestyle{选项}，页面样式，比如 empty 选项表示没有页眉和页脚。导言区还有其他全局性的设置等。

(4) 正文部分首先是文章标题\title{标题}。题目对于论文来讲是很关键的。常言说："题好文一半"。需要注意以下几点。

① 题目的长度适中，最好不要超过 20 个字。

② 题目的范围适中，选题过大或者过细都不合适。

③ 选择题目的一个标准是：读者通过题目就能知道这篇论文的主要研究内容。

通常在\title{标题}中输入标题，至于标题字体大小和位置(例如是否居中)，模板都已经设定好，无须作者设置。

(5) 作者信息\author{作者信息}。不同的模板，作者信息的格式不同，通常期刊或者会议给出的 LaTeX 模板中，都会给出示例。例如 LNCS 模板，如图 2.2 所示。

在源文件中，有几个地方需要注意。

① 如果作者多于一个，每个作者之间用\and 分隔开。

② 如果多个作者属于不同的单位，则可以 ^{1}，^{2}…表示不同单位的编号。

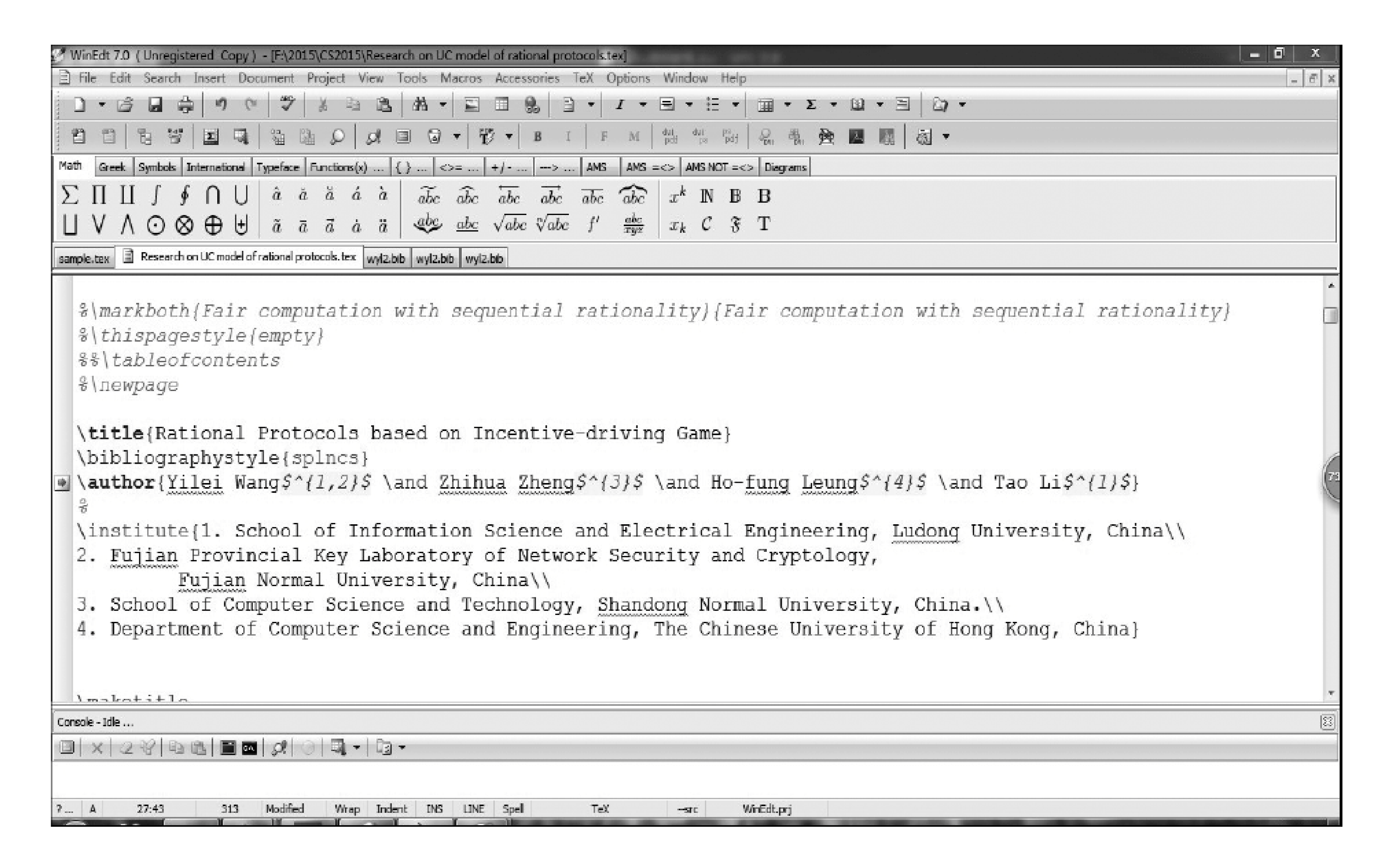


图 2.2　源文件中的作者信息

③ 还有更加复杂的情况，即同一个作者属于不同的单位，对于这种情况，就在该作者后面标注他所属的不同单位，如图 2.2 所示，第一作者属于第一和第二单位，则在该作者后面标注 ^{1,2}。

④ 作者单位使用\institute{…}来标注。

单击 pdftexify 编译该文件，得到的作者信息如图 2.3 所示。这里需要注意的是，pdftexify 是一个 WinEdt 自己写的脚本，相当于执行 pdflatex 然后再查看 PDF 文件。

Rational Protocols based on Incentive-driving Game

Yilei Wang[1,2], Zhihua Zheng[3], Ho-fung Leung[4], and Tao Li[1]

1. School of Information Science and Electrical Engineering, Ludong University, China
2. Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, China
3. School of Computer Science and Technology, Shandong Normal University, China.
4. Department of Computer Science and Engineering, The Chinese University of Hong Kong, China

图 2.3　编译后的作者信息

其中作者信息多行表示，用\\断行，自动居中。可以使用\maketitle表示本页为标题页，以便自动格式化。

(6) 接着是文章正文内容各部分了。摘要\begin{abstract}…\end{abstract}，或者直接\abstract。章节为\section{第一层标题}，\subsection{第二层标题}，\subsubsection{第三层标题}(注意：没有\subsubsubsection{第四层标题}这样的命令)。这几层编号在文章中对应的形式是1，1.1，1.1.1…。

(7) 致谢部分 \section*{Acknowledgements}。其中*表示该部分不计入编号。

(8) 再后面是参考文献部分，可以使用两种方法加入参考文献。第一种方法是手工逐条在正文尾部的\begin{thebibliography}{最大条数}…\end{thebibliography}内加入"\bibitem{关键词}文献信息"，文章中引用的地方用\cite{关键词}，自动按加入的顺序编号，形如[1]。第二种方法是使用bibTex。建立一个文献数据库文件：数据库名.bib，里面有按字段填写的文献信息，以及相应的"引用关键词"。bibTex会生成.bib文件，其中包含引用文献具体内容，在正文末尾用\bibliography{文献数据库名}包含该文件内容，注意文献数据库名不能包含空格。文章中引用格式同前面的方法，文献数据库中被引用的文献按格式出现在文末，未被引用的文献可以使用\nocite{关键词}来使其出现在文末。后一种方法的好处有：文献数据库可以共享；文献的内容与表现格式分离，内容填写更清晰，也可以更好地控制格式，比如文献的排列顺序。格式控制是在导言区加入\bibliographystyle{格式名}，其中格式包含在.bst文件里，可以是LaTeX提供的，也可以是期刊单位提供的。

从结构上看，文章算排版结束了。下面就文章正文内容的一些细节留些注意点。

(1) 插图：使用graphics宏包可以很方便地引用.eps格式图片，一般图片都是Matlab绘制图片，可以直接输出.eps格式。特别地，PS的.eps不行。图片一般集中放在当前目录下的子目录中，使用子目录在导言中用\graphicspath{{子目录名/}}，这个里面的{}不能少，图片文件名被引用时即可省略子目录名，指明.eps时效率高(第3章详细介绍图片的排版)。如果不指明.eps，就说明插入的图片是PDF格式。

(2) 表格：表格单元都是由内容撑起的，可以使用\rule[起始位置]{宽度}{高度}来撑起达到预期格式。rule定义的是一个矩形，起始位置指底线与当前行基准线的距离，负

值表示底线在基准线下面(第 4 章详细介绍表格的排版)。

(3) 插图、表格、公式都可以贴上各自自动编号的标签\label{关键词},引用时\ref{关键词}可以自动出现相应编号。

文章排版好了,输出文档需要注意几点。按照导言区格式设置,编译生成 DVI 作为中间预览基本不会有问题,但一般都需要最终 PDF 输出。

几个注意事项如下。

(1) 空格:LaTeX 中空格用来隔开单词(英语一类字母文字),多个空格等效于一个空格;对中文没有作用。

(2) 换行:用控制命令\\或 \newline。

(3) 分段:用控制命令\par 或空出一行。

(4) 换页:用控制命令\newpage 或\clearpage。

(5) 特殊控制字符:#、$、%、&、-、{、}、^、~,要想输出这些控制符用下列命令:\#、\$、\%、\&、\-、\{、\}、\^、\~。

2.2　各种 Package

LaTeX 中自带的宏包,有时无法满足一些排版、字体等方面的细节问题。例如图 2.4,如果把\usepackage{amsmath,amssymb}这一句注释掉(加符号%表示把该句注释掉,编译时不执行这一句),那么编译时会提示错误(Undefined control sequence)。这是因为在第 108 行有一个数学符号≈(该符号对应的命令为\thickapprox) 。这个数学符号需要用到宏包 amssymb,因此需要在源文件头部指明包含该宏包\usepackage{amsmath,amssymb}。

下面给出不同部分常用的宏包。

1. 页面与标题式样

1) geometry

利用 geometry 可以很方便地设置页面的大小。由于可以自动居中排放页面,自动计算并平衡页面各部分,如页眉、页脚、左右边空等的大小,因此只需给出很少的信息就能得

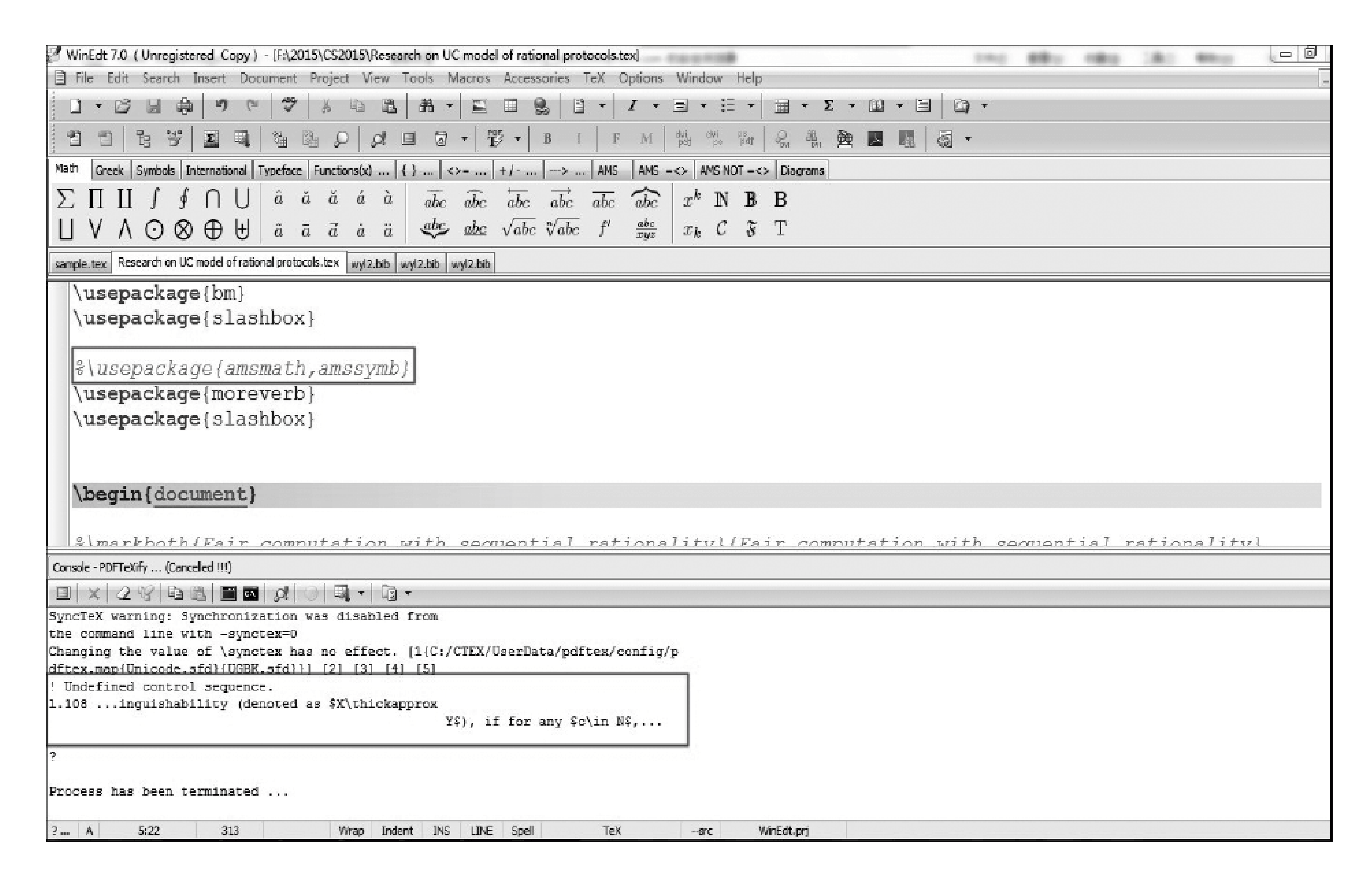


图 2.4　缺少宏包错误提示

到满意的页面。

2）rmpage

提供了简单的命令来设置页面的大小，并通过调整页面的宽度确保文本在打印区域内。如果页面需要特定的页面布局参数，最好还是使用上面的 geometry 宏包。

3）layout

显示文档的页面上各部分的设置。可用命令 \layout 来得到本文档的页面设置的视图。layout 是 LaTeX 标准的工具包 tools 之一，一般的 TeX 软件均包括此宏包。

4）layouts

比 layout 功能更强大，可显示文档的页面上各部分的设置。包括文本在一页中的位置，图表等浮动对象的位置移动，以及章节标题的设计及其在目录中的形式等。

5）multicol

提供了一个新的环境，使得可在一页上使用单栏和多栏版式。multicol 是 LaTeX 标准的工具包 tools 之一。一般的 TeX 软件均包括此宏包。

6）fancyhdr

用 fancyhdr 来设置页眉和页脚十分方便，而且可以在配合 CCT、CJK 来设置中文的

页眉等。

7）rplain

重新定义了 plain 页面样式，将页码放置在页面的左下角。在双面样式中，则分别为奇数页的左下角和偶数页的右下角。

8）pageno

可以将很方便地设置页码在页面上的放置位置。

9）titling

提供了一些命令用来控制由 \maketitle 命令生成的文档标题的样式。

10）titlesec

titlesec 可以让用户选择自己喜爱的标题样式，只需要几行简单的命令就可以实现。

11）sectsty

如同 titlesec 一样，提供了许多命令来使用户很方便地设计自己喜爱的章节标题的风格。

12）fncychap

另一个设计标题样式的宏包，主要是针对章的标题。

13）anysize

设定页面的大小，调整正文区和边空的大小。

14）crop

提供不同形式的截角标记，并提供选项来使排版的内容居中，标记垂直和水平的中轴线等。

15）fix2col

修补了标准的 LaTeX 双栏版式的一些不尽如人意的地方。

16）ragged2e

提供了一些新的命令和环境来协助 LaTeX 断词，从而尽可能地使排版得到的输出比较整齐。

17）scale

将整个的文档放大 1.44（\magstep2）倍。

2. 浮动对象及标题设计

1）floatflt

floatflt 宏包提供了 floatingfigure 和 floatingtable 两个环境,可将浮动图形或表格放置于文字段落的旁边。

2）float

利用该宏包可以定义自己喜欢的浮动对象的样式而不必拘泥于 LaTeX 所预定的设置。

3）rotating

可以将文本、表格、图形旋转,并提供了 sidewayfigure 和 sidewaystable 环境来使图形或表格横排。另外,也可以用 \rotcaption 命令来只对图形或表格的标题加以横排。

4）rotfloat

将 rotating 宏包和 float 宏包结合起来,通过对 float 宏包所定义的命令加以扩展,可以很方便地定义新的被旋转 90°或 270°的浮动对象。

5）endfloat

将所有的浮动对象放置于文章的最后分类排出。如将浮动图形都放置于文章的最后名为 Figures 的一章中,浮动表格等也类似地排放。

6）afterpage

提供命令 \afterpage,该命令使得所有作为其参数给出的 LaTeX 命令在当前页结束后才被执行。

7）placeins

提供 \FloatBarrier 命令,常用来解决过多未处理浮动图形的问题。

8）caption

提供了多种命令来更方便地设计浮动图形和表格的标题样式。

9）caption2

另一个功能强大的设计浮动对象的标题样式的宏包。

10）sidecap

轻松地得到标题在一边的浮动图形或表格。

11）fltpage

如果遇上图形或表格太大，以至无法和标题放置于同一页的情况，可以使用 fltpage 宏包。

12）subfigure

可以将一组图形或表格放在一个 figure 或 table 环境中，而每幅图形或表格都保持一定的独立性，可以有自己的标题等。例如，用户想把几幅图形分别编号为 Figure 1(a)，1(b)，1(c)，…，就可以用此宏包的\subfigure 命令来实现。另外，还提供\subtable 命令来处理表格的情况。

3. 生成与插入图形

1）LaTeX2e Graphics 宏包套件

LaTeX2e Graphics 宏包套件是 LaTeX 中插图所必备，是 LaTeX2e 所带的标准宏包。对不同的 DVI 驱动，提供了对 EPS、PS、PDF、TIFF、JPEG 等图形格式的支持。另外，该宏包还通过 color 宏包提供了对色彩的支持。

2）MetaPost

基于 MetaFont 的绘图语言。它的一些语法、命令等都和 MetaFont 类似，但不同的是它的输出为 PostScript，而不是位图。MetaPost 的绘图指令可以很好地融合在 TeX/LaTeX 文件中，在运行 TeX 或 LaTex 进行编译的过程中生成 PostScript 图形并插入到文档中。特别需要指出的是，尽管 PDFtex/PDFLaTeX 不支持 EPS、PS 格式的图形，但 MetaPost 的输出却可以很容易地在其中被使用。

3）PStricks

功能强大的绘图宏包，支持在 TeX/LaTeX 文件中直接使用 PostScript 命令，可以让用户在文档中轻而易举地得到各种 PostScript 的图形、文字效果。使用该宏包的文档需要用 dvips 等转换为 PS 文件后才能预览。另外，该宏包不能和 pdftex/pdflatex 配合使用。

4）XYpic

为在 TeX/LaTeX 文件中绘制 graph 和 diagrams 提供强大的支持。它可以和 Plain TeX、AMS-LaTeX、LaTeX，甚至 PDFTeX 一起配合使用。使用 XYpic 可以很方便地得

到各种曲线、箭头、多边形、直方图等。

5）psfrag

允许用 LaTeX 的文本和公式来替代 EPS 图形文件中的字符。在 CJK、CCT 等中文环境下，可以使用 psfrag 将图形中的标记字符替换所需的中文文本。

6）pspicture

使用 PostScript \special 重新实现了 LaTeX 的 picture 环境，使得可以设定任意角度和粗细的线段，对圆的大小也没有了限制。

7）texdraw

提供了许多命令来绘制各种样式的线段，bezier 曲线、圆、箭头等。也可以用不同的灰度来填充区域，在所绘制的图形上放置文本、数学符号需要 PostScript 的支持。

8）picins

picins 宏包定义了一个命令 \parpic，允许将图形等 LaTeX 对象放置在文本段落中。并且，设定适当的参数，可把该对象置于一带框的盒子、有阴影的盒子等。

9）picinpar

picinpar 宏包除了定义一个基本的环境 window，另外还有两个变体 figwindow 和 tabwindow。允许在文本段落中打开一个“窗口”，在其中放入图形、文字和表格等。

10）wrapfig

wrapfig 宏包提供了一个 wrapfigure 环境来排版窄小的图形，使得该图形位于文本的一边，并使文本在其边上折行。

11）eso-pic

可以很容易地在文档的每一页上都加上一幅或几幅图形。比较适合于用来得到水印效果。

12）overpic

允许直接将 LaTeX 对象放置到一幅图形上，而不是通过对图形上已有的标记进行替换来实现。overpic 宏包中定义了一个 overpic 环境，它有效地将 picture 环境和 \includegraphics 命令结合起来。使得 picture 环境的维数和插入的 EPS 图形的维数相同。这样就可以很容易地把 LaTeX 的命令放到图形上的任何指定位置。同时，还可以在

图形上加上标尺以方便定位。

13）epic 和 eepic

epic 提供了对 LaTeX picture 环境的有限的扩展，而 eepic 宏包则是在 epic 的基础上更进一步扩展了 LaTeX 的 picture 环境，使得可以画出任意角度的线段、任意大小的圆等。

14）trees

很容易画出任意大小的树形图。

15）curves

不需要太多的 TeX memory，就能得到各种具有连续角度的曲线，包括 bezrer 曲线、虚线等。

4. 表格与列表

1）array

增强了 tabular 环境的功能，可以更好地排版表格。

2）longtable

如果表格太长，超过了一页时，就可以试试 longtable 宏包所定义的 longtable 环境。

3）supertabular

自动计算表格的高度，把超出页面的表格部分放置在下一页。

4）tabularx

提供了新的表格环境 tabular * 、tabularx，可以设定表格的宽度。

5）ltxtable

简单地说，就是 longtable 和 tabularx 两个宏包的结合。

6）colortbl

利用该宏包可以设置表格中行、列等前景和背景色，从而得到彩色表格。

7）dcolumn

可以是用户将表格中的小数点对齐。

8）multirow

如果表格中某一单元横跨两行以上，就要用 multirow。

9）hhline

在表格中用 \hhline 得到的结果就如同 \hline 或 \hline\hline，当然在和垂直线的交叉处会有所不同。

10）slashbox

可在表格的单元格中画上一斜线。

11）booktabs

让表格中使用不同粗细的横线来划分行。

12）mdwtab

重新实现了标准的 LaTeX2e 的 array 和 tabular 的功能，并增加了新的内容。

13）paralist

提供新的列表环境，可以将 itemize 和 enumerate 列表排放在一段落中。

14）shortlst

专门用来排版列表项都很短的 itemize 和 enumerate 环境。

15）enumerate

给 enumerate 环境增加了一可选项，用来设定列表项的数字的形式。

16）multienum

支持将 enumerate 环境中的列表项用多列排出，即在一行中可以排出多个列表项。同时，提供了命令来设置每行中列表项的个数。

5. 目录与索引

1）tocloft

提供了让用户自己控制目录的样式的手段。

2）titletoc

设计自己喜欢的目录排版形式。

3）multitoc

允许在文档中只将目录(包括图形和表格目录)用两栏或多栏排版。

4）minitoc

使用该宏包可以将每一章的目录放置在该章的任何地方(一般在开始或结尾部分)。

5）tocbibind

使用该宏包可以将参考文献或索引等放置到目录中去。

6）shorttoc

使用该宏包可以在正式的目录前生成一个比较简略的目录，可以方便读者了解文档内容。这在排版比较大的书籍时很有用。

7）tocvsec2

该宏包可以控制出现在目录里每一章中编号的级别或是否给其编号。

8）makeindex

makeindex 不仅是一个 LaTeX 宏包，还有一个专门的同名应用程序来帮助生成 LaTeX 文档的索引。

9）nomencl

利用 makeindex 快速创建自己的符号命名列表。

6. 参考文献

1）bibtex

作为 LaTeX 的一个辅助程序，BibTeX 通过搜索一个或多个数据库，自动为 LaTeX 文档构造参考文献。

2）natbib

重新实现了 LaTeX 的 \cite 命令，使得既可以使用“作者-年代”形式的文献索引，也可使用通常的数字编号形式的文献索引。

3）footbib

定义了\footcite 命令，使得由该命令得到的参考文献的引用像脚注一样被放置在页面的底部。

4）custom-bib/makebst

利用标准的参考文献样式文件，设计自己的可供 BibTeX 使用的样式文件。

5）tocbibind

使用该宏包可以将参考文献或索引等放置到目录中去。

6）bibentry

使用该宏包可以在文本的任何地方放置参考文献的条目。

7）bibunits

使用该宏包允许文档的不同部分有各自的参考文献。这些部分可以是章、节或 bibunit 环境。

8）listbib

该宏包可以用来排版 BibTeX 的数据库文件，而且使用很少的 TeX 存储空间。这就使得可以排版很大的参考文献数据库文件。

9）gloss

使用该宏包可以借助于 BibTeX 创建文档尾部的注释表(glossary)。

10）mcite

使用该宏包可以在文中同时对多个参考文献的关键词进行引用。

11）varioref

该宏包定义了多个交叉引用命令，这些命令都是 LaTeX 的 \ref 命令加上一些文本后得到的，而这些加上的文本可以很方便地被替换为不同的语言，如中文等。

12）fancyref

该宏包的引用命令 \fref 可以根据标记的前缀给出不同的引用文字。比如 \fref{eq：first} 会给出“Equation (1) on page 2”，而 \fref{sec：first} 则会给出“Section 1 on page 2”。当然，这些前缀和文本的形式用户都可以自己来设定。

13）prettyref

该宏包为 LaTeX 的交叉引用机制提供了附加功能，使得用户可以预先设置所有类型的标记(label)和 fancyref 的功能差不多。

7. 数学与化学公式

1）AMS-LaTeX

作为 AMS-LaTeX 在 LaTeX 中的实现，AMS-LaTeX 包括两部分：一部分是 amsmath 宏包，其主要目的是用来排版数学符号和公式，其中专门有 amsthm 宏包，提供对定理的排版；另一部分是 amscls，提供了美国数学学会要求的论文和书籍的格式。

2）AMS Fonts

美国数学学会还提供一套的数学符号的字库，这套字库中增加了很多 TeX 的标准字库 Computer Modern 所没有的一些数学符号，如粗体数学符号等。

3）theorem

通过定义不同的 theorem 环境，自己定义定理、定义、引理等的样式。

4）subeqn

提供了 subequations 和 subeqnarray 环境，可以对数学公式中的子式进行编号，得到如(1a)、(1b)、(1c) 这样的公式编号。

5）subeqnarray

定义了 subeqnarray 和 subeqnarray＊ 环境，可对一组公式中的每行进行编号，给出如(1a)、(1b)、(1c)等的编号。

6）mathenv

提供了几个很有用的数学命令和环境，可以得到比相应的标准的 LaTeX 命令或环境更好的排版结果。

7）eqnarray

定义了 equationarray 环境，将 LaTeX 标准的 eqnarray 环境和 array 环境结合起来。

8）youngtab

定义两个命令来排版如 Young-Tableaux 的式子。

9）yhmath

提供了一系列很大的分界符，如( )、< >、[ ] 等。

10）tmmath

支持用 Adobe Times 和 TM-Math 字族来排版文本和数学公式。

11）vector

提供了一组新的数学命令来排版各种样式的向量。

12）nicefrac

在正文文本中排版分式时，可以用它来得到较好的排版效果。

13）mdwmath

定义了\sqrt * 命令来得到没有上面的横线的根式符号，此外还定义了其他一些数学符号。

14）Bold math symbols

定义了\bm 命令，可用来得到加粗的斜体字体。

15）ntheorem

扩展了 LaTeX theorem 环境的功能，并解决了设置定理环境的结束标记的问题。

16）easybmat

排版块状矩阵。可以设置相同宽度的列，或等高的行，或两者同时设定。此外，还可以在行或列之间加上各种直线。

17）harpoon

提供了一些命令在文本上方或下方加上带有半个箭头的线段标记。

18）chemsym

由 Mats Dahlgren 设计，目的在于正确地排版化学元素的名称。它提供了 109 条相应于化学元素的命令，其命令名称与元素的化学符号完全一致。

19）xymtex

Shinsaku Fujita 在 1993 年到 1995 年期间开发的专门用于绘制化学中有机分子等结构的一组宏，它由一组 LaTeX 宏包组成。

20）ppchtex

ppchtex 是 ConTeXt 中的独立模块，专门用来排版化学符号和公式。

8. 抄录和代码打印

1）verbatim

重新实现了 LaTeX 的 verbatim 和 verbatim * 环境，并提供了新的环境 comment 和 verbatiminput 来在文档中加入评论和直接抄录文件。一般的 TeX 软件均包括此宏包，是 LaTeX 标准的工具包 tools 之一。

2）moreverb

应用上面的 verbatim 宏包所提供的命令，对抄录环境进一步加以扩展。主要是增加

了与制表符有关的一些功能,将抄录的内容写入一文件以备重复使用等。

3）fancyvrb 与 fvrb-ex

fancyvrb 宏包提供了方便的命令来设计不同样式的抄录环境。如使用不同的字体、颜色,加入行号、边框等。还可根据不同的条件对抄录的文本使用不同的样式。fvrb-ex 宏包则利用 fanvyvrb 所提供的命令给出了一个 example 环境,允许在列出包含 TeX 命令的文本的同时将该文本排版。

4）sverb

提供 list 等环境,可将抄录环境中的内容写入外部文件中,也可从外部文件中读入。

5）listings

排版 C、C++ 、Pascal 等源代码,提供语法加亮显示的功能。

6）algorithms

提供排版算法步骤的 algorithmc 和 algorithm 环境,对其中的关键词可采用不同的显示效果。

7）newalg

定义了排版算法步骤的 algorithm 环境。

8）program

排版编程语言的源代码或算法步骤。

9. 特殊文本元素

1）footmisc

提供了许多命令来弥补标准的 LaTeX2e 中\footnote 命令的不足,包括可以用符号替代脚注的数字编号,将脚注放置在边注区,在同一地方使用多个脚注等。

2）footnote

改进了标准的 LaTeX2e 的\footnote 命令,使得可以在\parbox、minipage 和 table 环境中标记的脚注能够被正确地放置在整个页面的下方脚注区中。

3）ftnright

使用这个宏包可以在多栏版式的文档中,将一页上的所有脚注都放置在最右边一栏的底部,而不是放置在各自所在栏的底部。

4）footnpag

自动设置脚注的计数器，使得对每一页上的脚注都可以设置自己的编号。这里的编号不仅仅是数字，也可以是其他符号。该宏包可以很好地配合标准的 LaTeX2e 文档类。

5）savefnmark

可以将 table 或 minipage 环境中的脚注加以标记，并可在后面再次使用。

6）abstract

可以用来方便地设置 abstract 环境，特别是当在双栏版式中排版单栏的简介时。

7）lastpage

将标记 Lastpage 写入.aux 文件中，允许使用者引用文档的最后一页。比如在页脚可以得到 Page 2 of ××× pages 这样的效果，这里的“×××”就是用 \pageref{Lastpage} 得到的文档页码总数。

8）xr

利用此宏包的 \externaldocument 命令，可以实现对外部文档的标记的引用。

9）hyperref

扩展了 LaTeX 的所有的交叉引用的命令（包括目录、参考文献等）的功能，使其生成各种驱动如 dvips、pdftex 等可识别的\special 命令，从而得到超文本链接。此外，该宏包还提供了新的命令来支持在文档中加入对外部文档和 Internet 网址的链接。

10）schedule

排版时间表的宏包。

11）acronym

提供了很多命令帮助用户在文档中方便地处理首字母缩略词，并在文档的最后生成一个列表。

12）hypbmsec

扩展了\section 命令，允许在\section 命令中同时给出出现在标签（Bookmarks）和正文中的标题，而这些标题可以有所不同。这是因为出现在\section 命令中的标题不一定符合 PDF 标签的要求，如不能使用 TeX 命令等。

13）hyphenat

可以在文档中取消 TeX 自动断词的功能，也可以在某一单词后再恢复这一功能。

14）units

基于 nicefrac 宏包，提供对计量单位比较美观的排版效果。

15）slunits

提供对国际标准的计量单位符号的支持。

16）soul

支持对单词加上下划线或其每个字母在一定的宽度内均匀散布。

17）altfont

使用该宏包，可以在一个宏包中使用多种不同的字体，包括 PSNFSS 和 MFNFSS。

18）prelim2e

可以在每页页脚下方标记出本文档的版本信息等。

19）lineno

在每行文本前加上行号，并且可以用 LaTeX 交叉引用来引用它们。

20）moresize

重新定义了\huge 命令，此外还定义了比其更大的字体。

21）texshade

texshade 是一个 TeX/LaTeX 多序列排序、比较的软件，它可以用同一种颜色标记出各个序列中相同部分。当然，它的功能不会只有这一点。

22）niceframe

定义了\niceframe 等新的命令，可以将文本等放置在用 dingbat 字体生成的装饰框内。

23）fancybox

提供了\shadowbox、\doublebox、\ovalbox 和\Ovalbox 4 个命令来生成不同形状的盒子。

24）indentfirst

让每一章节开始的段落也缩进。可以和标准的 LaTeX 文档类配合使用。

10. 辅助工具包

1) pagesel

利用此宏包可以很方便地从输出页面中选取一页或多页。

2) count1to

用 page、part、…、subparagraph 的值设置计数器 \count1、…、\count8，而 \count9 则用来标记奇数页。通过显示这些计数器的值并将其写入 .dvi 文件中，可以实现有选择地打印文档的某一部分。

3) stdclsdv

对正确识别 LaTeX 的标准类文件中所提供的章节的级别，如为有无 \chapter 或 \section 这一级的命令提供了一个解决办法。

4) showlabels

帮助用户跟踪所有的标记(labels)。每当使用 \label 命令或遇到一个自动编号的公式时，就把新的标记的名字放置在所在页的边空中。

5) showkeys

该宏包修改了 \label、\ref、\pageref、\cite 和 \bibitem 命令，使得这些命令所使用的"内部标记"被显示在边注区或其所得结果的上方，并且尽可能不影响排版的结果，为标准的 LaTeX2e 工具包之一。

6) fileerr

定义了多个文件使得可以很容易地从找不到文件的错误循环中退出，为标准的 LaTeX2e 工具包之一。

7) calc

重新实现了 LaTeX 的命令 \setcounter、\setlength、\addtocounter 和 \addtolength。使得可以在这些命令里使用符号表达式，为标准的 LaTeX2e 工具包之一。

8) changebar

通过在页边空处加上一竖直条来标记 LaTeX 文档中改动过的部分等。

9) alphalph

提供了两个命令\alphalph 和\AlphAlph，可将数字转换为字母。

10）typehtml

使用此宏包可以在 LaTeX 文档中处理 HTML 代码。

11. 非标准文档样式

1）seminar

不经意间就做成了令人满意的投影胶片。

2）foiltex

排版幻灯片、胶片，并且可以和 fancybox 配合得到很好的立体效果。

3）pdfslide

排版幻灯片、胶片。配合上 hyperref，用 pdflatex 编译生成 PDF 并经 ppower4 处理后，可以得到与 PowerPoint 相媲美的演示效果。

4）pdfscreen

不用复杂的命令就能设计并得到精美的 PDF 文档。可以让文本显示在不同形状的窗口中，再加上导航按钮、背景。

5）texpower

排版可以在屏幕上演示的投影片。它允许使用 PStricks、XYpic 等 pdflatex 所不支持的宏包，但需要用 Acrobat Distiller 来得到最后的 PDF 文件。

6）KOMA-Script Class

这是一套按照欧洲的排版标准设计的 LaTeX2e 的文档类。与 LaTeX2e 所提供的标准文档类稍有不同。如果 CJK 中使用中文的章节号，需要使用这套文档类。

7）geom

以标准的 LaTeX 的 article 类和 book 类为基础，增加了许多功能。

8）exam

用来排版试题的文档类。

9）draftcopy

在文档的某些页面印上 DRAFT 字样的水印。

10）labels

用来制作地址标签。

11）AMS Book/Paper

美国数学学会的书稿和论文的样式文件。

12）paper/journal

paper/journal 是对 article 类的扩充，定义了几个有用的命令来增强对标题和关键词等的处理。

13）a0poster

提供了特大号的字体，可以排版 A0 纸大小的海报。

2.3 页面设置

通常情况下，页面的格式由 cls 文件配置，用户无须自己设定。但有时需要用户自己进行页面设置，例如毕业论文模板，有些学校提供了模板，但有些页面需要自己设置，这时就需要用户自己根据需要设置页面格式。主要的页面设置包括以下几个方面。

1. 页面大小

如果用户确实需要自己设置页面大小，可以使用 2.3 节中的宏包，也可以不用加载任何宏包就可以很容易设置纸张大小，只需在 documentclass 加入关于纸张大小的选项即可。系统默认的纸张大小是 A4：

```
\documentclass[a4paper]{article}
```

其他可用的选项还有：

(1) a4paper (297mm * 210mm)。

(2) a5paper (210mm * 148mm)。

(3) b5paper (250mm * 176mm)。

(4) letterpaper (11in * 8.5in)。

(5) legalpaper (14in * 8.5in)。

(6) executivepaper (10.5in * 7.25in)。

这些页面设置，实际上是在设置页面的高度\paperheight 和宽度 \paperwidth。如果

需要其他页面的高度,可以手动更改它们的值。

```
\setlength\paperheight{高度}
\setlength\paperwidth{宽度}
```

2. 页边距和页眉页脚

在 2.3 节中提到 geometry 宏包,使用 geometry 宏包,可以让页边距和页眉页脚的设置变得非常简单。

```
\documentclass[a4paper]{article}
 \usepackage{geometry}
 \geometry{left=2.5cm,right=2.5cm,top=2.5cm,bottom=2.5cm}
 \begin{document}
        Test
 \end{document}
```

常用的长度选项还有 head、headsep 和 foot,如图 2.5 所示。

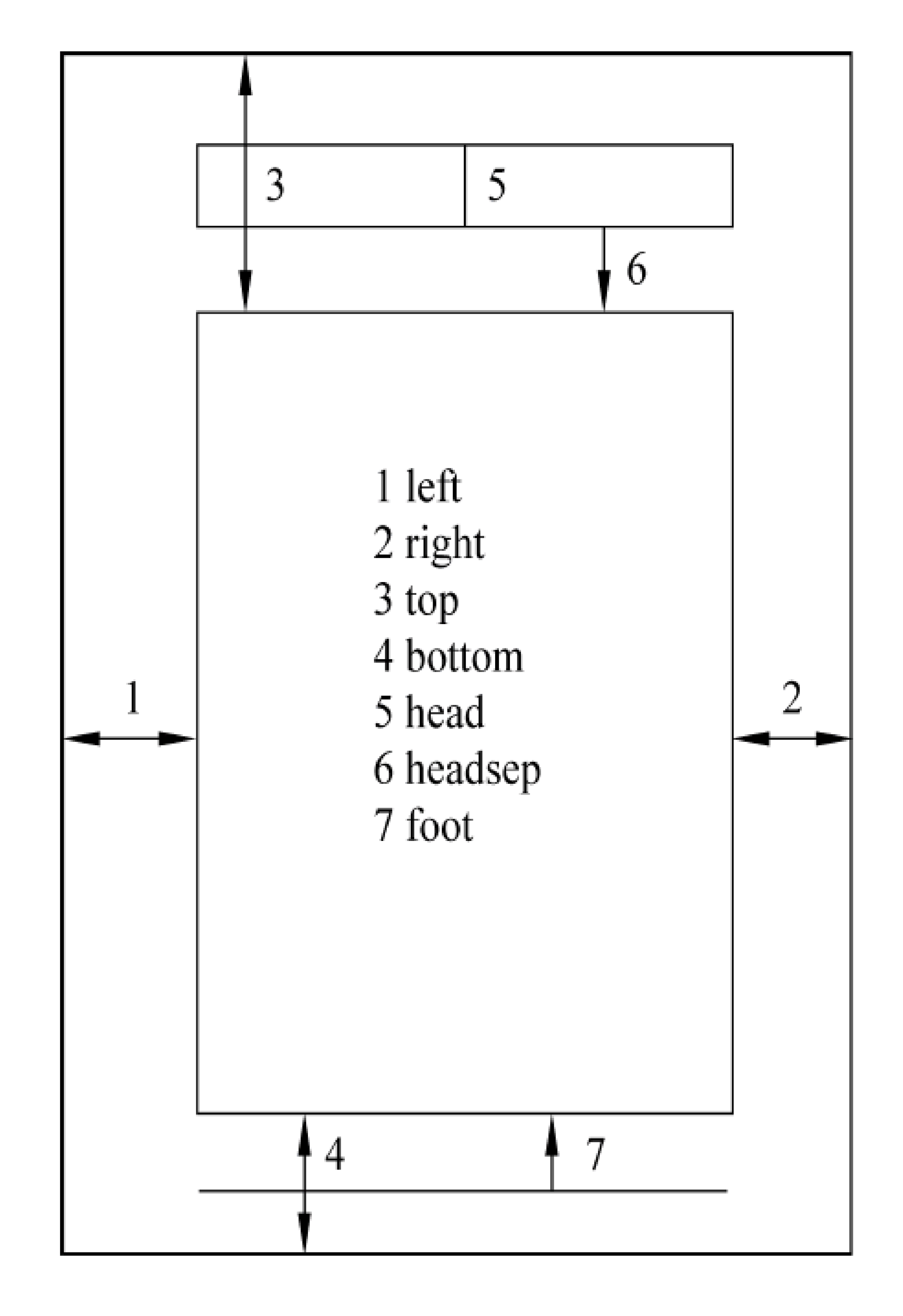


图 2.5　LaTeX 页边距和页眉页脚

3. 常见段落格式设定

(1) 字间距(Letter-spacing)是指一组字母之间相互间隔的距离。字间距影响了一行或者一个段落的文字的密度。在 LaTeX 中可以使用\renewcommand {\CJKglue} {\hskip 宽度} 命令调节字间距。\hskip 用来设置水平空距,\baselineskip 表示当前字间距。需要注意的是,该命令只适用于 CJK 和 xeCJK。例如:

```
\renewcommand{\CJKglue}{\hskip 1pt plus 0.08\baselineskip}
```

(2) 行间距是指从一行文字的底部到另一行文字底部的间距。通常行间距设置为一倍行间距,但是根据不同的需要也可以设置不同的行间距。使用间距宏包\usepackage {setspace},然后在需要设置行间距的内容之间加上如下命令行:

```
\begin{spacing}{行距 }
        ⋮
    \end{spacing}
```

例如,如果想把行间距调整为 2 倍行间距,则使用:

```
\begin{spacing}{2.0}
        ⋮
    \end{spacing}
```

一倍行距的内容如图 2.6 所示,两倍行距的内容如图 2.7 所示。

(3) 段间距是指两段之间的距离,可以设置\parskip 的值,比如\setlength {\parskip}{0.5\baselineskip}。

(4) 首行缩进,通常 LaTeX 模板第一段不缩进,如果没有特别要求,就可以按照模板排版。如果希望每一行都可以首行缩进,就可以使用 indentfirst 宏包进行设置。根据前面的要求,需要在 tex 文件的前部加上宏包\usepackage{indentfirst},加上宏包之后,如果需要指定某段首行缩进,在段首加 \indent;如果指定某段首行不缩进,在段首加 \noindent。另外,除了默认的缩进量以外,用户还可以自己设定缩进量,使用语句 \setlength\parindent{2em}即可。

(5) 居中、左对齐、右对齐。

Abstract. Rational computation means secure multi-party computation in the presence of rational parties, who care about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain actions in the computation. Therefore rational parties may carefully choose their actions before they interact with others. Traditionally, the main tasks of rational computation is to encourage parties to exchange information with their opponents, even if they do not trust each other. Exchanging information is similar to the action of cooperation towards the view of game theory. Therefore, measures should be taken to encourage parties to cooperate with others in rational computation. In this paper, we assume that rational parties who participate in the computation protocol form a social cloud. Parties in the social cloud may interact within several rounds and in each round, some desirable properties such as reputation may be generated. Parties who have good reputation means they are likely to cooperate with others. The structure of social cloud is not static. Instead, it evolutes when parties complete one round of computation. We mainly discuss the impact of social cloud structure on rational computation. Simulation results show that when the community structure reach to a proper value, parties are more likely to cooperate in the computation protocol. In addition, rational parties can gain optimal utilities.

图 2.6　一倍行距

Abstract. Rational computation means secure multi-party computation in the presence of rational parties, who care about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain actions in the computation. Therefore rational parties may carefully choose their actions before they interact with others. Traditionally, the main tasks of rational computation is to encourage parties to exchange information with their opponents, even if they do not trust each other. Exchanging information is similar to the action of cooperation towards the view of game theory. Therefore, measures should be taken to encourage parties to cooperate with others in rational computation. In this paper, we assume that rational parties who participate in the computation protocol form a social cloud. Parties in the social cloud may interact within several rounds and in each round, some desirable properties such as reputation may be generated. Parties who have good reputation means they are likely to cooperate with others.

图 2.7　两倍行距

设置文档的对齐有两种形式：一种是环境形式 center、flushleft、flushright；另一种是命令形式 centering、raggedright、raggedleft。与命令形式不同，环境形式会插入新段落，如果不希望插入新段落，就需要使用命令形式。命令形式通常配合环境使用，限制有效范围在环境内。还要注意的是，与 flushleft 对应的是 raggedright，与 flushright 对应的是 raggedleft，不要弄反了。

如果想让一段文字居中，可以在源代码中使用下列命令行：

```
\begin{center}
Rational computation means secure multi-party computation in the presence of rational parties, who care about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain actions in the computation.
\end{center}
```

或者下列命令行：

```
\begin{flushleft}
Rational computation means secure multi-party computation in the presence of rational parties, who care about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain actions in the computation.
\end{flushleft}
```

文字居中效果如图 2.8 所示。

Rational computation means secure multi-party computation in the presence of rational parties, who care about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain actions in the computation.

图 2.8　文字居中效果

左对齐所对应的命令行如下：

```
\begin{flushleft}
Rational computation means secure multi-party computation in the presence of rational parties, who care about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain actions in the computation.
\end{flushleft}
```

或者如下：

```
\begin{flushleft}
Rational computation means secure multi-party computation in the presence of rational parties, who care
about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain
actions in the computation.
\end{flushleft}
```

文字左对齐效果如图 2.9 所示。

Rational computation means secure multi-party computation in the presence of rational parties, who care about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain actions in the computation.

图 2.9　文字左对齐效果

右对齐的命令行如下：

```
\noindent
\begin{minipage}{\linewidth}
\raggedright
Rational computation means secure multi-party computation in the presence of rational parties, who care
about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain
actions in the computation.
\end{minipage}
```

或者使用 \begin{flushright}…\end{flushright}，命令行如下：

```
\begin{flushright}
Rational computation means secure multi-party computation in the presence of rational parties, who care
about maximizing their utility. Here the notion of utility denotes payoffs when parties take certain
actions in the computation.
\end{flushright}
```

文字右对齐效果如图 2.10 所示。

注意：默认的情况是两端对齐，所以一般不对文字的对齐效果做设置，如果有特殊要求，可以按照上述命令行进行设置。

图 2.10　文字右对齐效果

（6）其他页面设置。除了上述常用的一些页面设置以外，有关页面距离的设置如下所示：

\columnsep：列间距。

\topmargin：页眉到页边的距离。

\topskip：页眉与正文的距离。

\textheight：正文的高度。

\textwidth：文本的宽度。

\oddsidemargin：奇数页的左面页边距。

\evensidemargin ：偶数页的左面页边距。

有关段落缩进的页面设置如下：

\parindent：段落缩进距离。

\parskip：段落间的距离。

\floatsep：浮动对象之间的距离。

\textfloatsep：最后一个浮动对象顶端或第一个浮动对象底端与正文之间的距离。

\intextsep ：文中浮动顶端与底端所留的距离。

\dbltextfloatsep：在双列输出时用 \textfloatsep 的数值。

\dblfloatsep：在双列输出时用 \floatsep 的数值。

\abovecaptionskip：标题上方的距离。

\belowcaptionskip：标题下方的距离。

有关数学公式的页面设置如下：

\abovedisplayskip：公式前的距离。

\belowdisplayskip：公式后面的距离。

\arraycolsep：在一个 array 中列之间的空白长度。

有关列表的页面设置如下：

\topsep：第一个 item 和前面版落间的距离。

\partopsep：当在一个新页开始时加到\topsep 的额外空间。

\itemsep：连续 items 之间的距离。

第3章　图形排版

科技论文除了文字部分，还包括图形和表格，通常用图形的方法说明实验的效果会更加直观。相对于文字部分，LaTeX 中图形的排版更加复杂。与 Word 中图形的所见即所得的情况不同，LaTeX 中图形的排版是通过命令行实现的，图形的位置、大小等属性在编译之后才能看到效果。另外，LaTeX 对于图形的文件格式也有要求，必要时还需要进行转换。对于多个图片的排版更加复杂，本章描述了图形排版中出现的一些问题。

3.1　图形格式

在 LaTeX 中，最常用的图片格式是 EPS 和 PDF 格式。

3.1.1　EPS 格式

EPS 称为被封装的 PostScript 格式，又称为带有预视图像的 PS 格式，它是由一个 PostScript 语言的文本文件和一个(可选)低分辨率的由 PICT 或 TIFF 格式描述的代表像组成。EPS 文件是目前桌面印刷系统普遍使用的通用交换格式当中的一种综合格式。EPS 格式的缺点是文件通常相当大，而且除非使用 PostScript 打印机，否则只能印出低分辨率的预视档或根本印不出图形。如果在不支持 PostScript 的输出设备上输出 EPS 文档，唯一的方法是先使用 RIP(Raster Image Processor)程序将其转换成位图格式，再打印。

1. 使用 Adobe Photoshop 生成 EPS 格式的文件

如果图形是在 Adobe Photoshop 中完成的，那么可以按照以下步骤生成一个 EPS 格式的文件。当然对于一些模拟数据产生的图形，例如使用 MATLAB 产生的图形，可以先保存为 JPG 格式的图片，然后使用 Adobe Photoshop 打开该图片。

(1) 打开需要保存为 EPS 图片的文件,如图 3.1 所示。

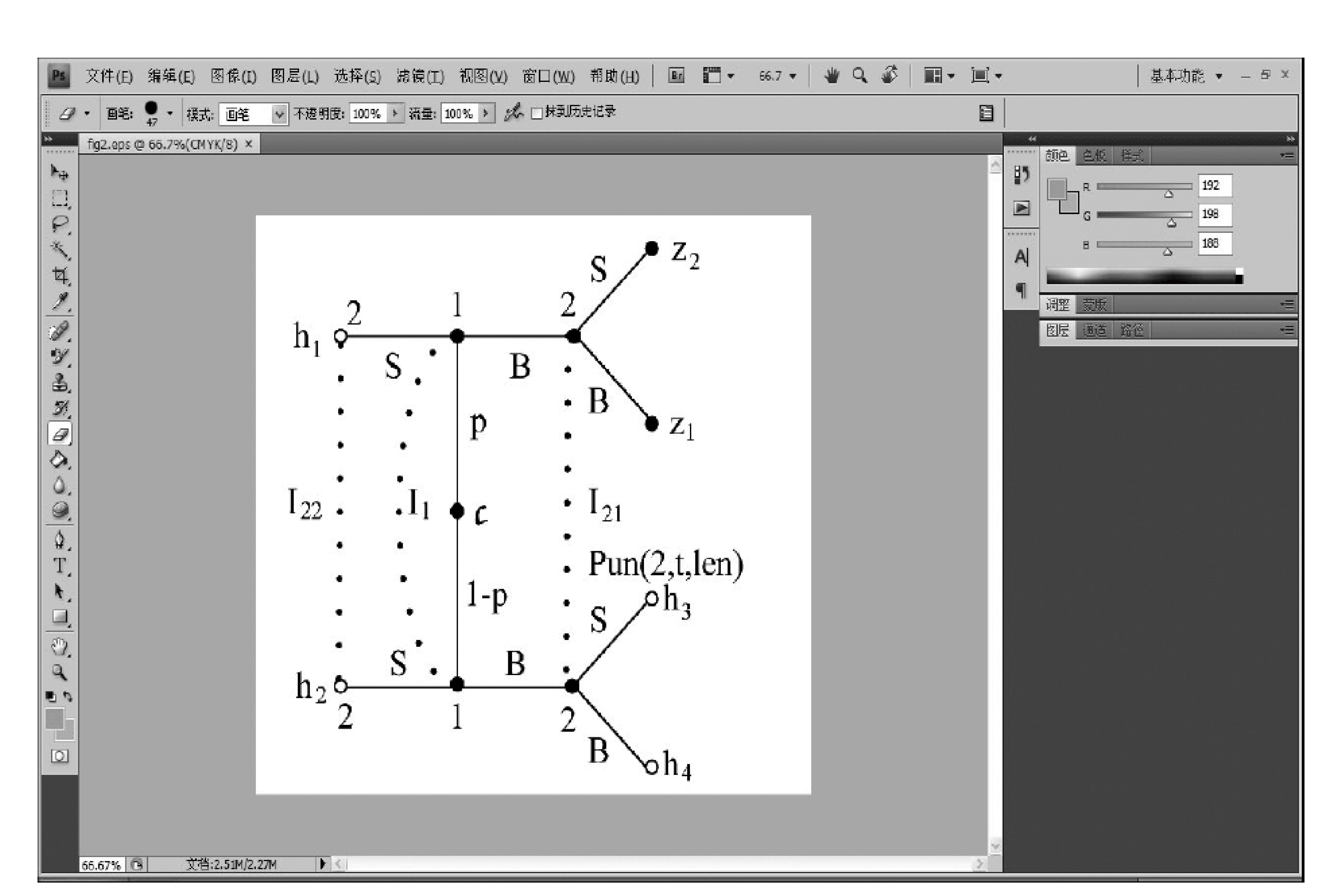


图 3.1　用 Adobe Photoshop 打开图片

(2) 单击“文件”菜单按钮。

(3) 在打开的“文件”菜单中选择“存储为”选项。

(4) 弹出“存储为”对话框,在“保存在”下拉列表框中选择要保存的图片位置。

(5) 在“文件名”后的文本框中填写文件名。

(6) 在“格式”下拉列表中选择 EPS,如图 3.2 所示。文件名为 fig2.eps。

(7) 单击“保存”按钮。

保存以后,在对应的文件夹中,会显示已经保存好的 EPS 文件,如图 3.3 所示。

此时如果直接将 fig2.eps 插入到 LaTeX 中,会发现图形周围的白边也被保留下来,为了避免这种情况发生,需要进一步地对 fig2.eps 进行处理。

(1) 双击文件 fig2.eps,使用 GSview 打开该文件,如图 3.4 所示。

单击 OK 按钮,显示图片,如图 3.5 所示。

(2) 选择 file→PS to EPS 命令,弹出如图 3.6 所示的界面。勾选 Automatically calculate Bounding Box 复选框,系统自动将图片周围白边裁剪掉。再插入 fig2.eps 会发

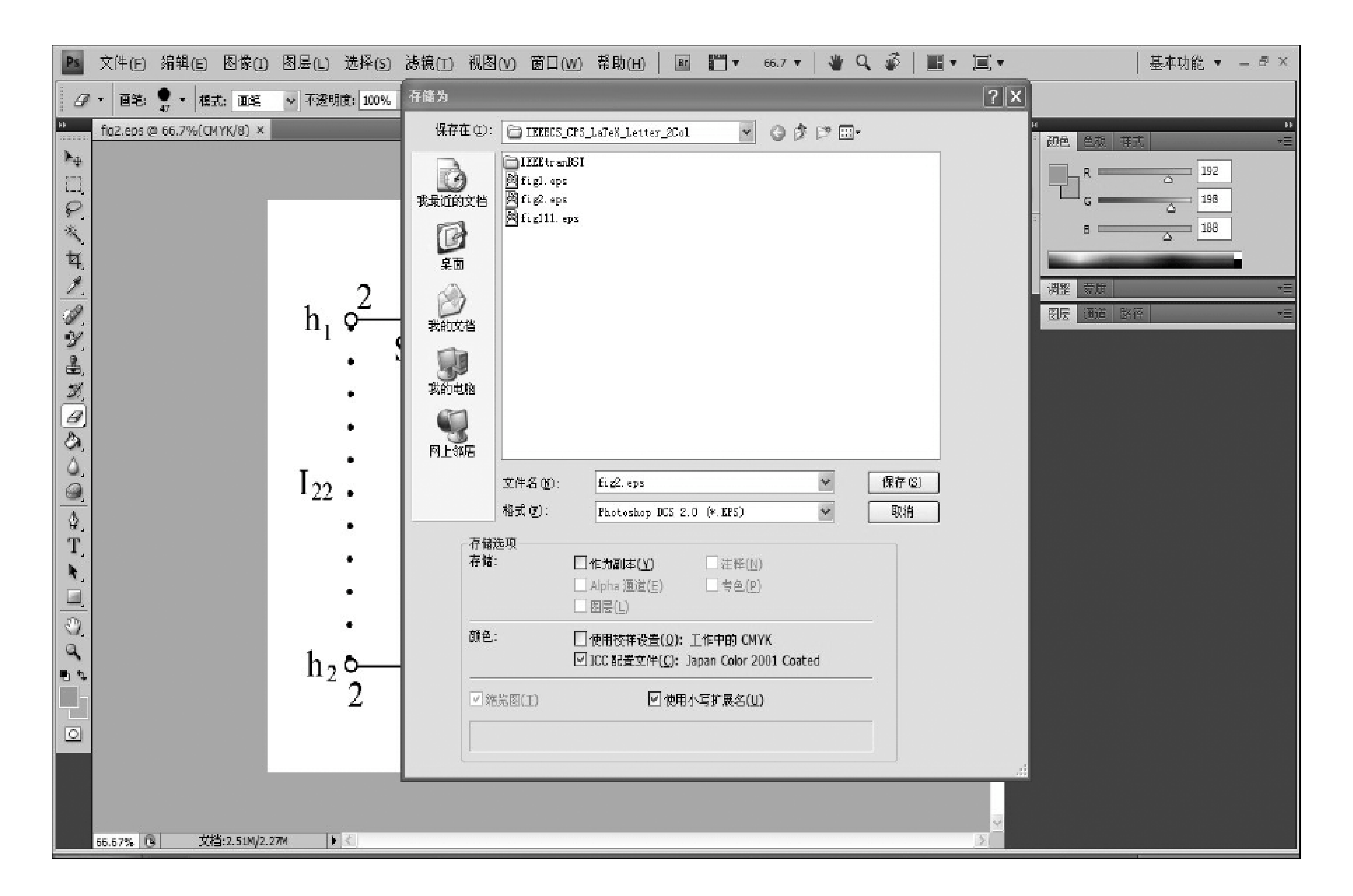


图 3.2　用 Adobe Photoshop 保存图片



图 3.3　保存图片的图标

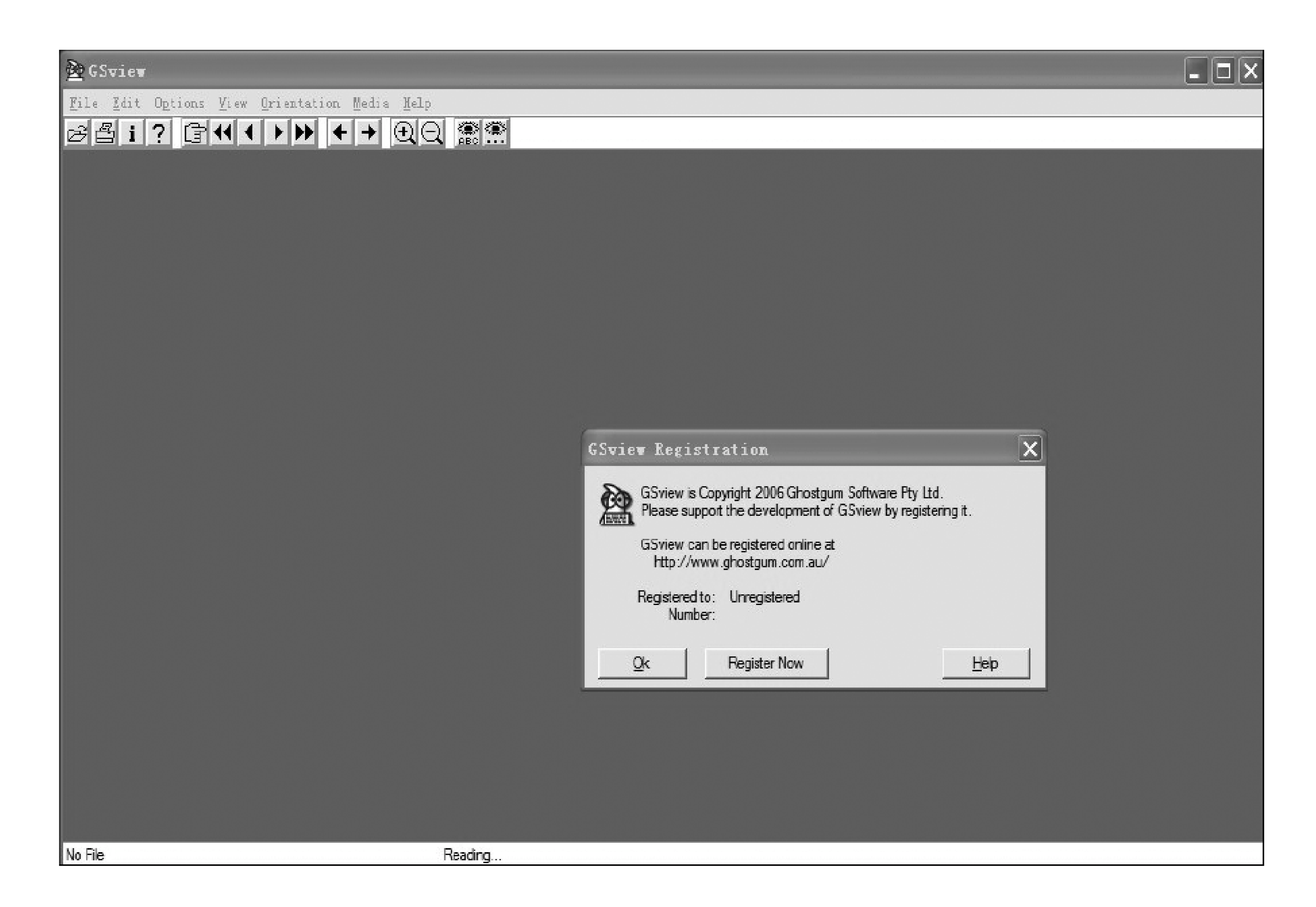


图 3.4　用 GSview 打开图片

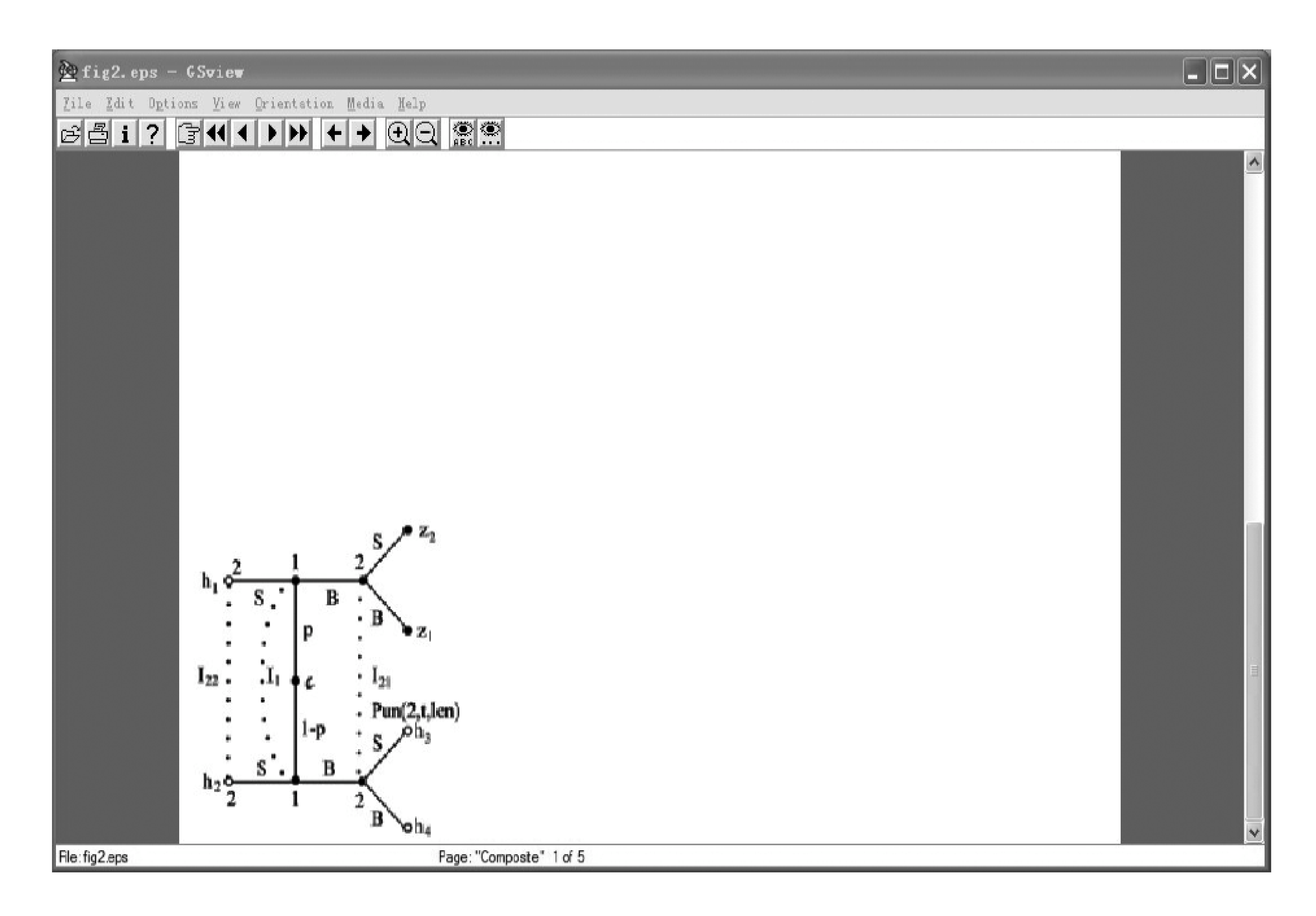


图 3.5　GSview 显示图片

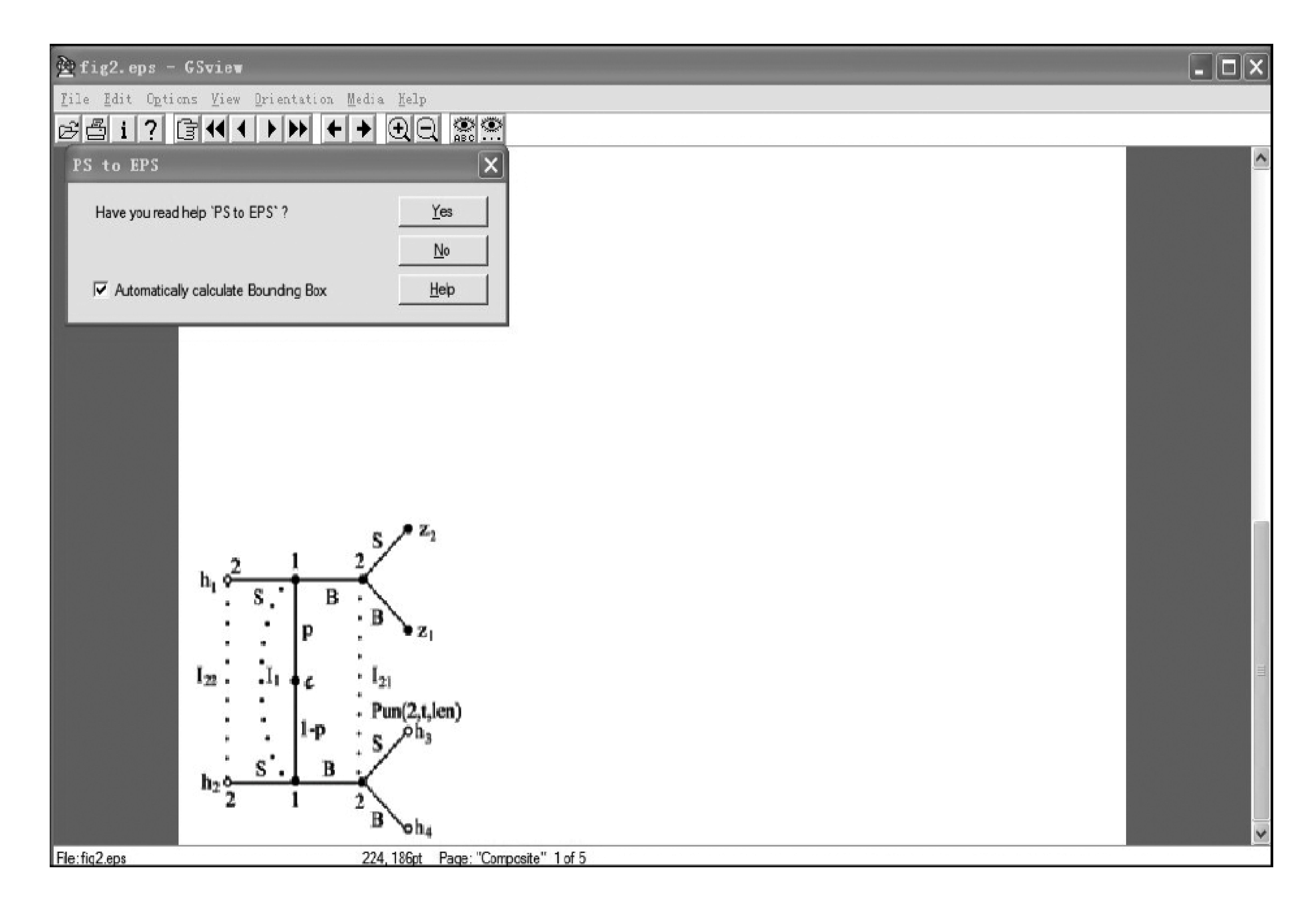


图 3.6　用 GSview 自动裁边

现,周围的白边被消除了。

2. 使用 Visio 生成 EPS 文件

除了使用 Adobe Photoshop 画图以外,一些类似流程图或者网络拓扑图,使用 Visio 画图更加清晰。然而 Visio 却不能直接生成 EPS 格式的文件,所以只好借助于第三方软件。

第一种方法是通过 Visio 另存为 *.emf 格式然后用 openoffice 的 Draw 打开,可以导出为 EPS 格式。导出时需要选中打开的图片,这样导出 EPS 之后就只包含 *.emf 图片的内容,不会有大片的空白。但是用这个方法导出的 EPS 图片会走样,比如直线经常会断断续续的。

第二种方法是通过 Acrobat 打印机将 Visio 图形打印成 PDF,再经过 GSview (GSview 4.8)裁剪,导出的 EPS 文件十分完美。具体过程如下。

(1) 编辑图片文件,保存为 PDF 文件。在这一步需要首先安装 Acrobat,在 Visio 和 Word 下把编辑好的图片打印为 PDF 文件即可,如图 3.7 所示。

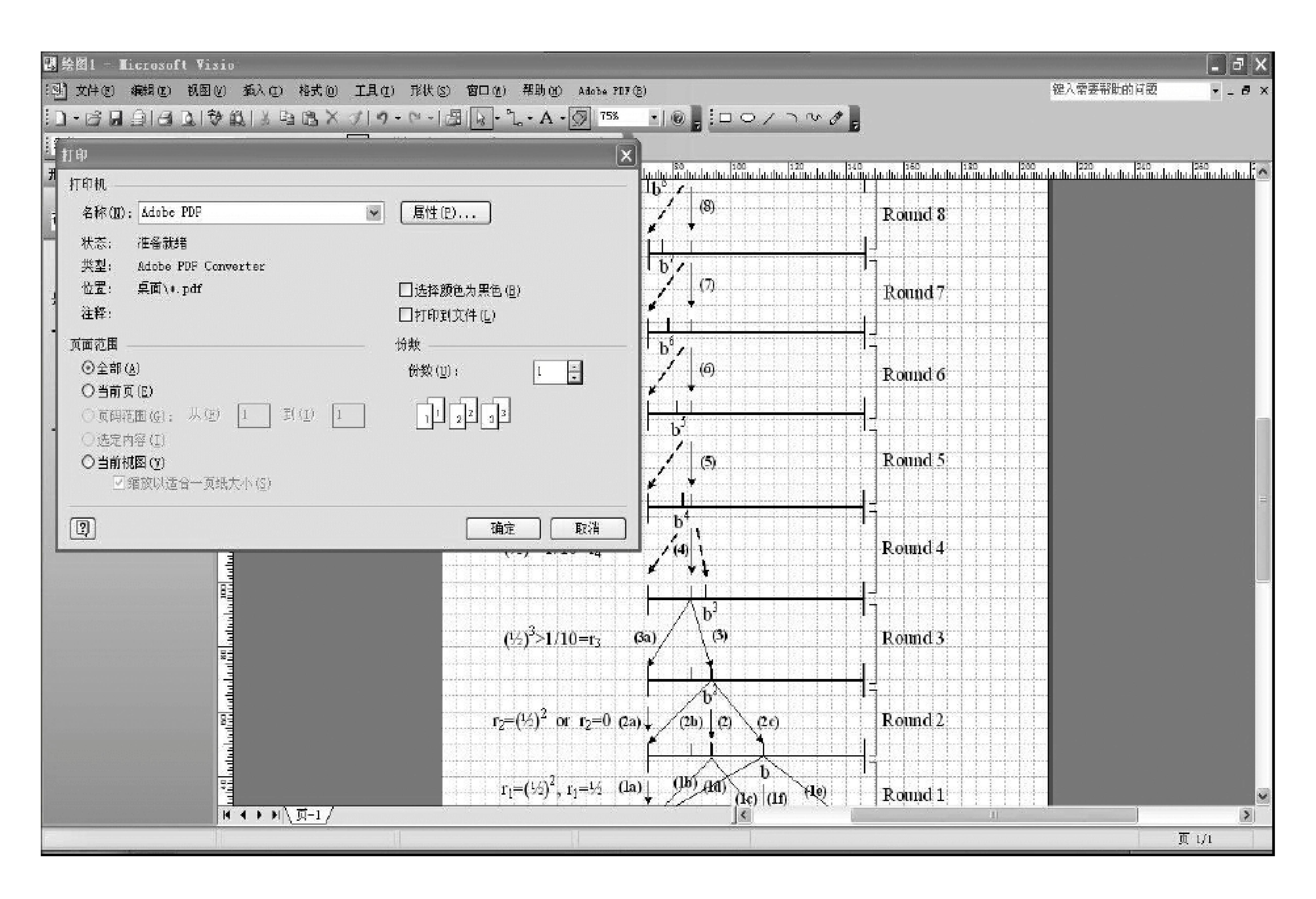


图 3.7 Visio 转化成 PDF

(2) PDF 文件转换为 EPS 文件。在 Acobat 下打开 PDF 文件,选择另存为 EPS 文件

即可，如图 3.8 所示。

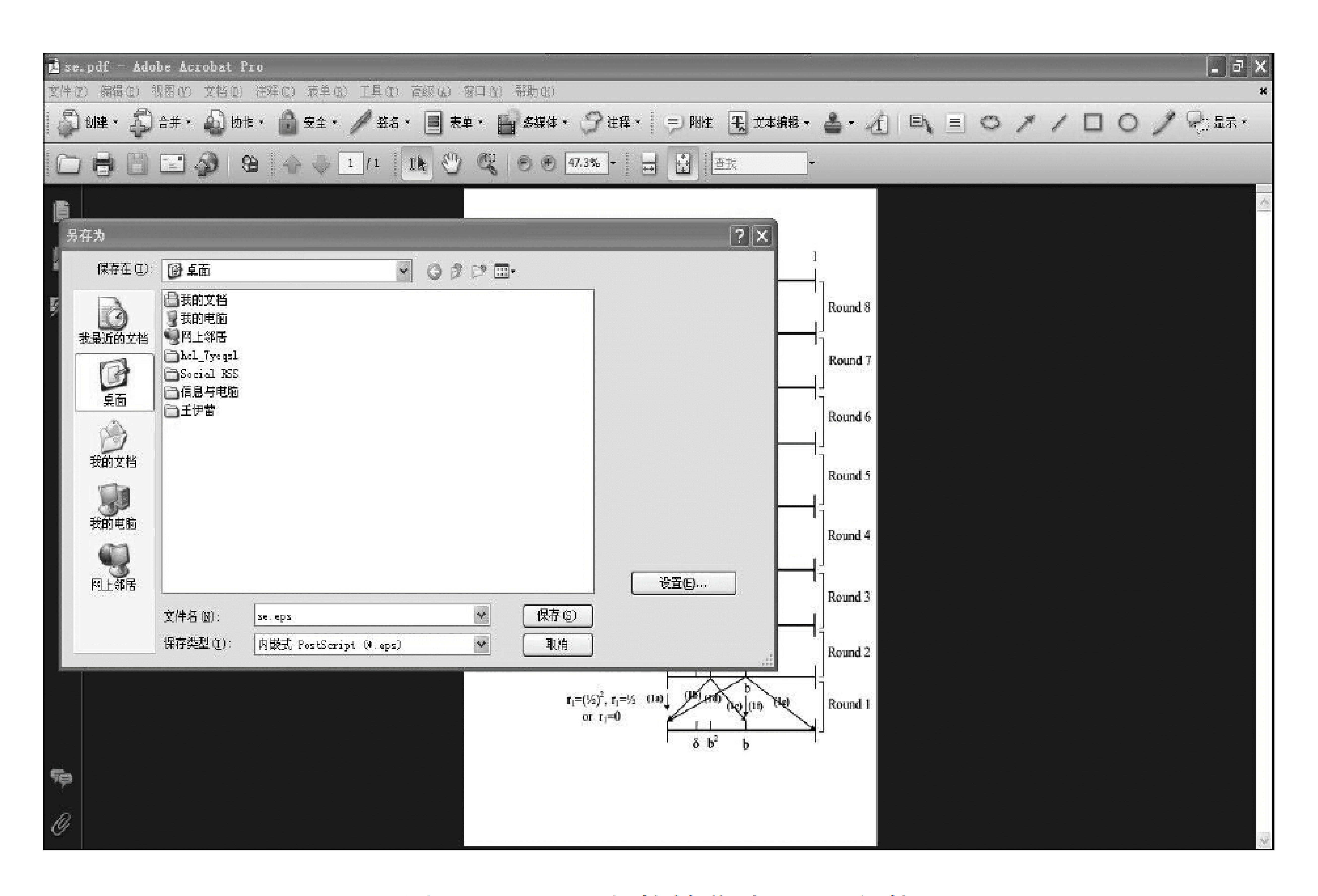


图 3.8　PDF 文件转化成 EPS 文件

（3）对 EPS 文件裁剪。用 GSview 打开 EPS 文件，选择 File→PS to EPS 命令，选择自动裁剪，存为另一个 *.eps 文件即可，如图 3.4～图 3.6 所示。

3.1.2　PDF 格式

使用 WinEdit 编辑 LaTeX 源文件时，可以直接插入 PDF 格式的图片。使用 PDF 格式的图片比 EPS 格式的图片简单，不需要转换。可以直接从 Visio 或者 Adobe Photoshop 生成 PDF 文件。但是转换后也存在一个问题，就是生成的图片会有一些白边，需要裁掉。如果图片是 Visio 生成的，就可以利用图 3.7 将图片转换成 PDF 格式。然后使用 Adobe Acrobat Pro 打开 PDF 文件，如图 3.9 所示。

从图 3.9 可以看出，图片周围有大片留白，如果此时将图片插入到 LaTeX 中，图片周围这些空白会影响图片的显示。可以使用 Adobe Acrobat Pro 中的裁剪工具“工具”→“高级编辑”→“裁剪工具”命令将留白去掉，如图 3.10 所示。

裁剪后，将图片保存，再次插入 LaTeX 文件中，会发现留白不见了，如图 3.11 所示。

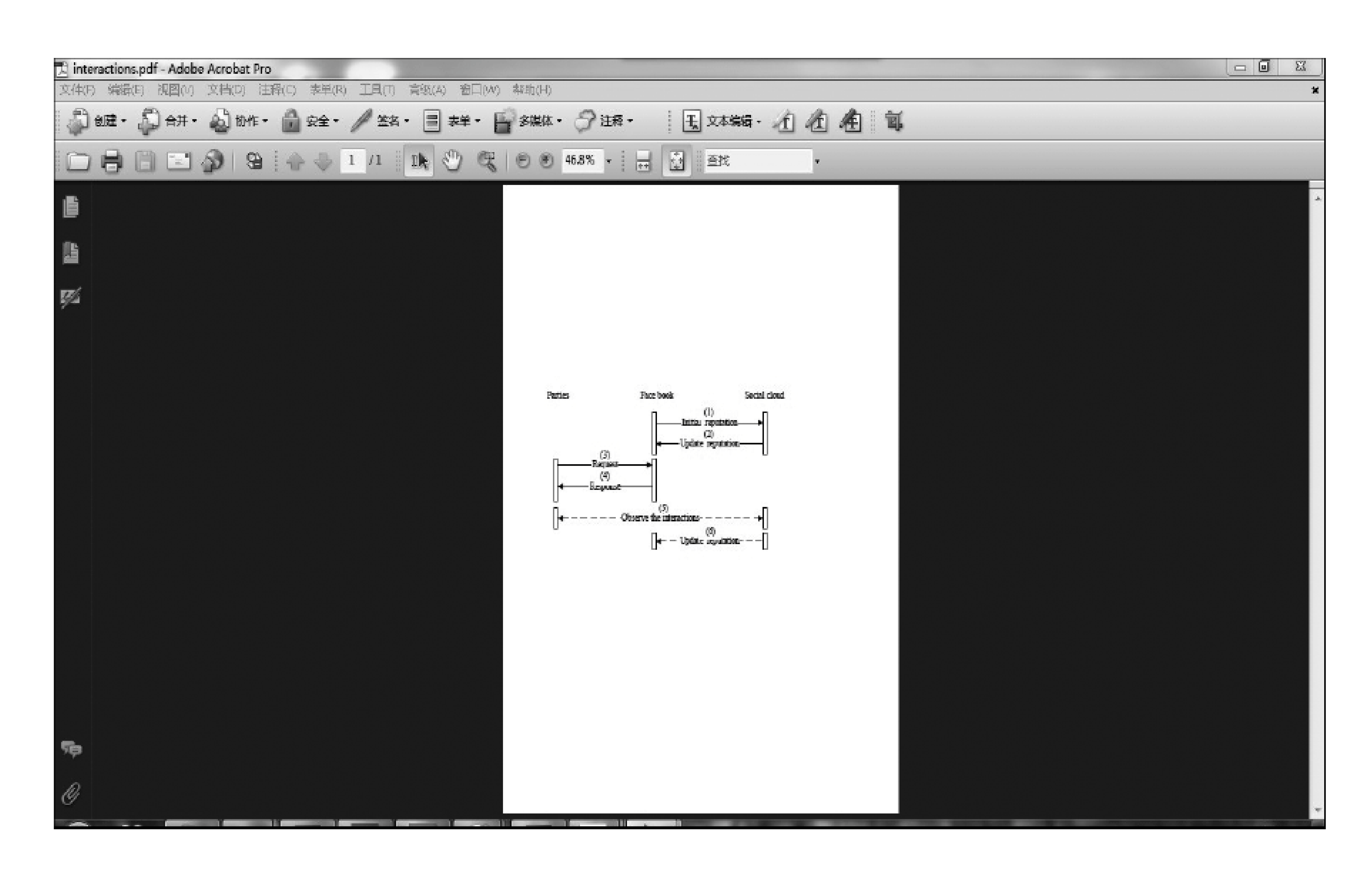


图 3.9　Visio 转换成 PDF 文件

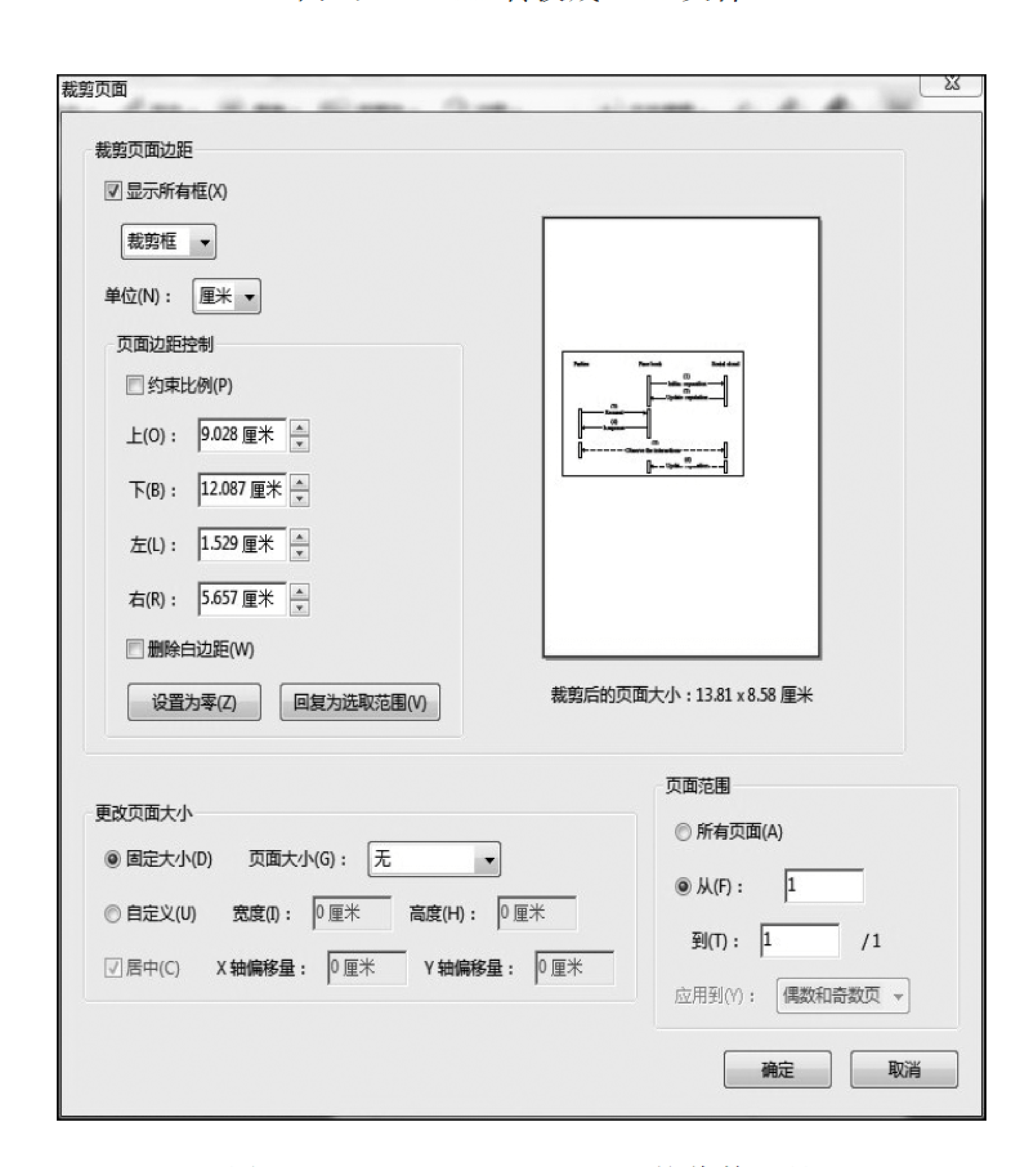


图 3.10　Adobe Acrobat Pro 的裁剪工具

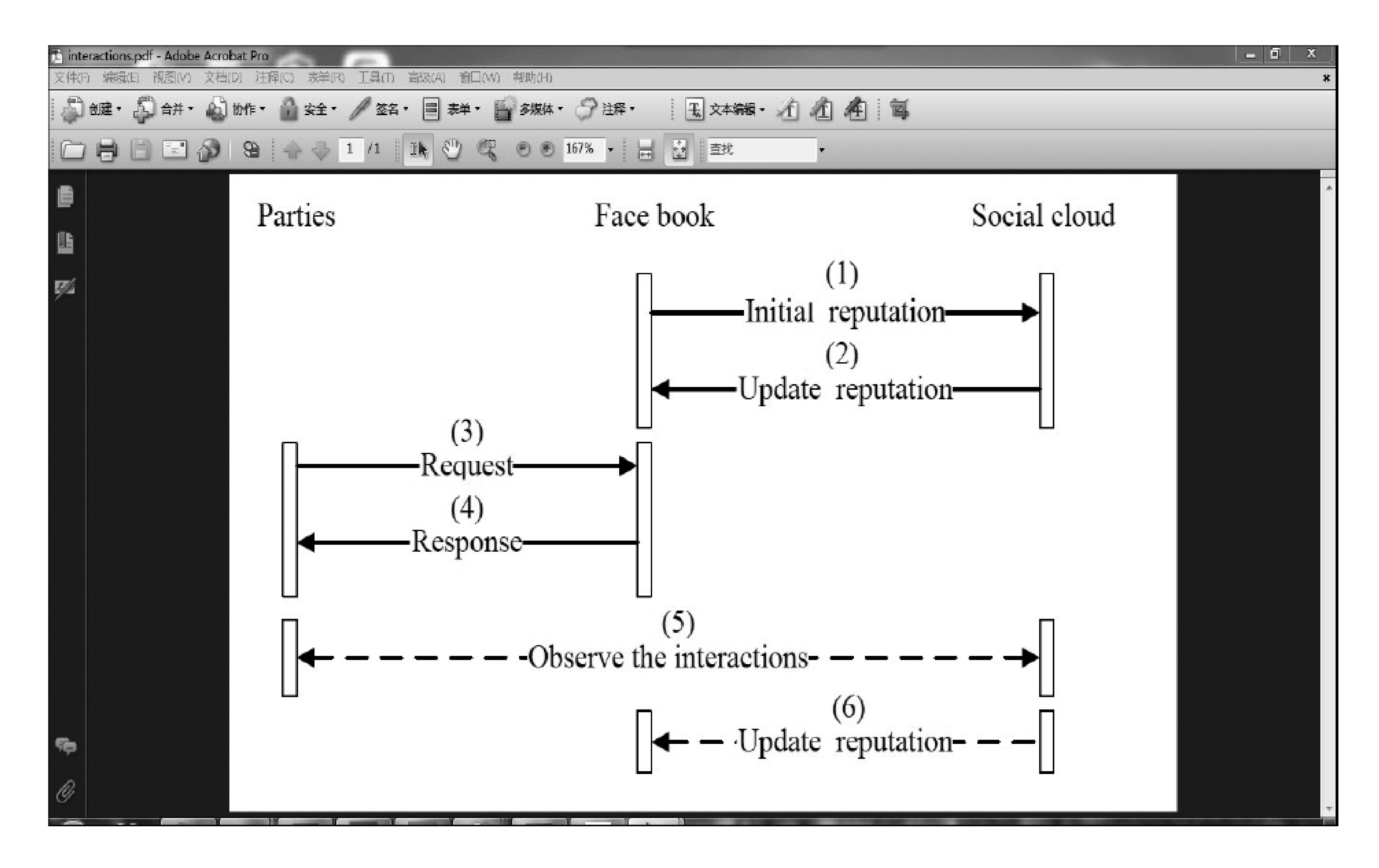


图 3.11　图片裁剪后的效果

3.2　图形属性设置

在 Word 中,图片的位置需要用户自己设置,在 LaTeX 中,可以通过命令行设定图片的位置。插入图形的基本命令行如图 3.12 所示。

```
\begin{figure}[!htp]
\centering
\includegraphics{×××}
\caption{title}
\label{fig:example}
\end{figure}
```

图 3.12　插入图形的基本命令行

(1) \begin{figure}…\end{figure}表示插入一个图片。

(2) [!htp]设置图片的位置,详细说明参见 3.2.1 节。

(3) \centering 表示图片居中显示。

(4) \includegraphics{×××}表示插入图片,×××表示图片的文件名,建议用户把图片和 TEX 文件放在一个文件夹下,这样×××只标明文件名即可。如果图片比较多,可以单独为这些图片建立一个文件夹,在引用这些文件时,需要指出文件所在的文件夹位置。

(5) 在\caption{title}中,title 表明图片的标题。

(6) \label{fig:example}中的 fig:example 表明图片在引用时的标识。

3.2.1 图形位置设置

图形(figure)环境有一个可选参数项允许用户来指示图形有可能被放置的位置。这一可选参数项可以是下列字母的任意组合。

(1) h:当前位置。将图形放置在正文文本中给出该图形环境的地方。如果本页所剩的页面不够,这一参数将不起作用。

(2) t:顶部。将图形放置在页面的顶部。

(3) b:底部。将图形放置在页面的底部。

(4) p:浮动页。将图形放置在一个允许有浮动对象的页面上。

注意:

① 如果在图形环境中没有给出上述任一参数,则默认为[tbp]。

② 给出参数的顺序不会影响最后的结果。因为在考虑这些参数时,LaTeX 总是尝试以 h-t-b-p 的顺序来确定图形的位置。所以[hb]和[bh]都使 LaTeX 以 h-b 的顺序来排版。

③ 给出的参数越多,LaTeX 的排版结果就会越好。[htbp]、[tbp]、[htp]、[tp] 这些组合得到的效果会更好。

④ 编译时不能使用 pdflatex,会出错。即使不出错,也看不到图。应使用 latex 编译生成 dvi,然后 dvi2ps、ps2pdf 就可以看到图了。

⑤ 只给出单个的参数项极易引发问题。如果该图形不适合所指定的位置,它就会被搁置并阻碍对后面的图形的处理。一旦这些阻塞的图形数目超过了 18 幅(这是 LaTeX 所能容许的最大值),就会产生 Too Many Unprocessed Floats 的错误。

当 LaTeX 试图放置一浮动图形时,它将遵循以下规则。

(1) 图形只能置于由位置参数所确定的地点。

(2) 图形的放置不能造成超过版面的错误。

(3) 图形只能置于当前页或后面的页中。所以图形只能“向后浮动”而 不能“向前浮动”。

(4) 图形必须按顺序出现。这样只有当前面的图形都被放置好之后才能被放置。

① 只要前面有未被处理的图形,另一幅图形就不会被放在当前位置。

② 一幅“不可能放置”的图形将阻碍它后面的图形的放置。直到文件结束或达到 LaTeX 的浮动限制。

同样地,另一表格也只能在其前面的表格都被处理完后才能被放置。不过,表格在排版时是跳过图形而单独处理的。

3.2.2　图形大小设置

在 Word 中可以双击图片设置图片的大小,或者可以直接拖动图片,更改图片的大小。在 LaTeX 中也可以通过命令行来设置图片的大小。

在/includegraphics 中可以加入[选项]来指定图片大小:

```
/includegraphics[width=3in]{file.eps}
```

设定图片宽度为 3 inches,图片高度会自动缩放。

```
/includegraphics[width=/testwidth]{file.eps}
```

除了直接设定图片宽度外,还可以设定图片比例。

```
/includegraphics[width=0.8/textwidth]{file.eps}
```

设定图片宽度为文本宽度的 0.8 倍。

```
/includegraphics[width=/testwidth-2.0in]{file.eps}
```

设定图片宽度比文本宽度少 2 inches。

也可以使用[选项]指定图片旋转角度:

```
/includegraphics[angle=270]{file.eps}
```

将图片旋转 270°。

两个选项同时使用,中间用逗号隔开:

```
/includegraphics[width=/testwidth, angle=270]{file.eps}
```

例如，对于图片 fig2.eps，希望能够旋转 270°，并且图片宽度是 2.5inches。对应的命令行如图 3.13 所示，效果图如图 3.14 所示。

```
\begin{figure}[!ht]
\centering
\includegraphics[angle=270,width=3in]{interactions}
\caption{The interactions of rational computation based on social cloud.}
\label{interactions}
\end{figure}
```

图 3.13　设置图片大小命令行

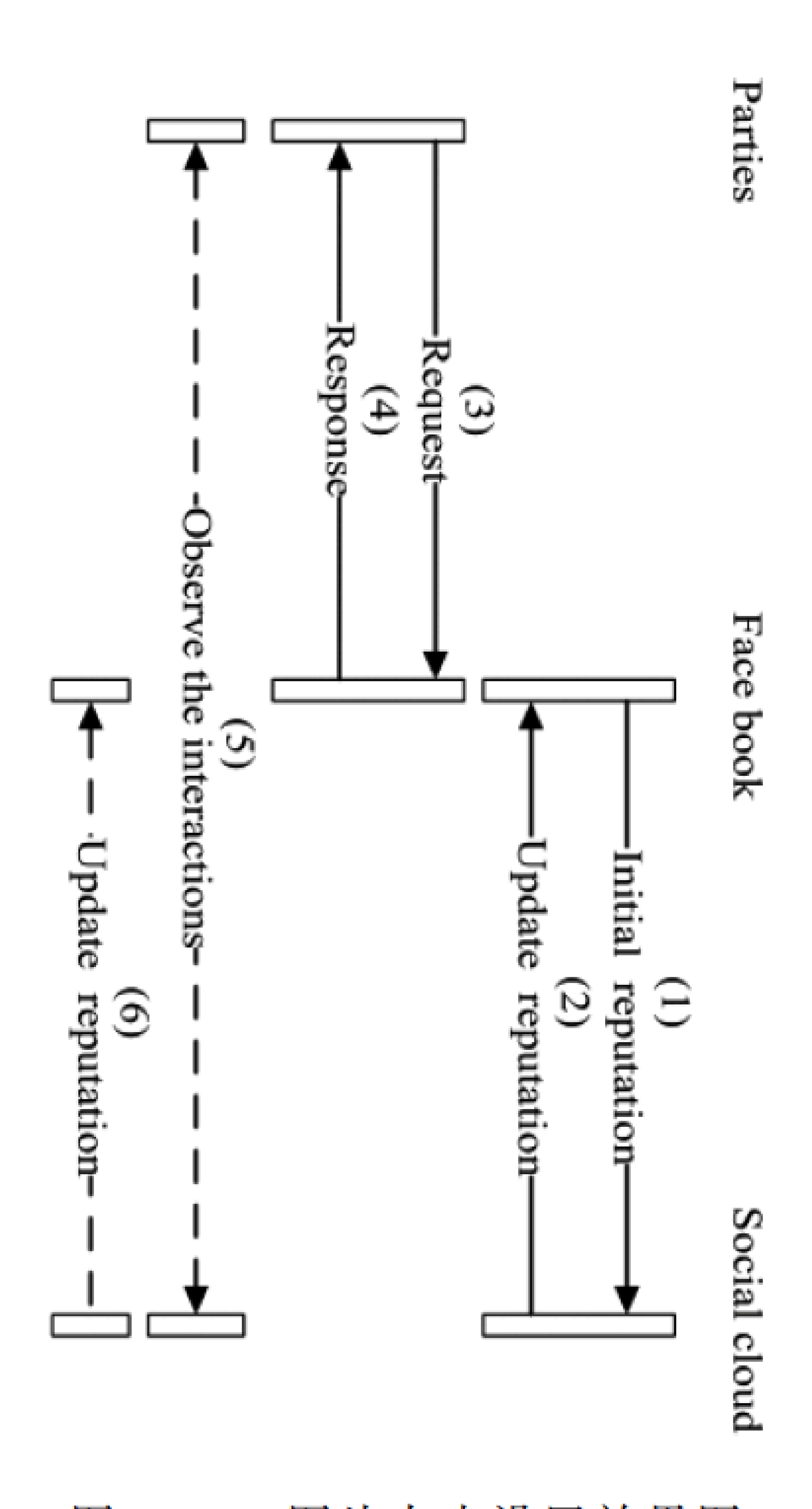


图 3.14　图片大小设置效果图

在 LaTeX 文档中插入图片都是通过使用一些 LaTeX 图形处理宏命令来实现的，有很多宏命令都支持在 LaTeX 文档中插入 eps 格式的图形文件，主要步骤如下。

(1) 在 LaTeX 文档的文件说明部分加上 graphicx 包，使用语句 /usepackage {graphicx}。

(2) 在正文中用 includegraphics 宏命令引用图片，具体语句如下：

```
/includegraphics[height=高度]{图片文件名}
```

或者：

```
/includegraphics[width=宽度]{图片文件名}
```

(3) 然后在需要插入图片的地方使用语句。

其中的“高度”和“宽度”是指希望图片打印的高度和宽度，必须给出单位，可用厘米(cm)或英寸(in)。高度和宽度也可用上述格式同时给出，这样可以改变原图的长宽比例。上述命令中的图片文件名是指欲插入的图片文件的文件名，图片必须是 eps 格式的。

用 graphicx 包的 includegraphics 宏命令插入图片时还可以使图片旋转，方法如下：

```
/includegraphics[height=高度][angle=旋转角度]{图片文件名}
```

插入的图形通常为 eps、pdf 或者 jpg、png 等格式，假设名字为 fig，把它放在 tex 文档同一目录下。

① 先在导言区加一句/usepackage{graphicx}。

② 如果想插入 inline 的图形，直接使用/includegraphics[width＝5in]{fig}，不用加后缀名。

这个 width 是最常用的选项，也可以改成其他的。

③ 如果想插入浮动图形，使用如下命令：

```
/begin{figure}[htbp]
/centering/includegraphics[width=3.5in]{fig}
/caption{something}
/label{fig:1}
/end{figure}
```

④ 如果是 eps 的图形，编译过程是 latex、dvips、ps2pdf。如果是 jpg，则导用下面的包：

```
\usepackage{graphicx}
```

然后在文章中想要加入的位置放置图片，格式([ ]中的内容是可选的，{}中是图片和

后缀名)如下：

```
\includegraphics[width=1.77in,height=1.75in]{pic.jpg}
```

然后用 PDF LaTeX 编译即可。

这是一个简单的添加图片的例子，如果要添加浮动图片，则要用到如下命令：

```
\begin{figure}
\centering
\includegraphics[width=1.77in,height=1.75in]{pic.jpg}
\caption{This is an inserted JPG graphic}
\label{fig:graph}
\end{figure}
```

如果是 pdf、png 图形，编译过程同 jpg，即 PDFLaTeX。用 includegraphics 宏命令(graphics 包)。

引用图片还可以使用 psfig 宏命令。

(1) 用 psfig 宏命令。

首先需在 LaTeX 文档的文件说明部分加上：

```
\usepackage{psfig}
```

然后在需要插入图片的地方引用：\psfig{figure=图片文件名，height=高度} 或者 \psfig{figure=图片文件名，width=宽度}。

其中的“高度”和“宽度”是指希望图片打印的高度和宽度，必须给出单位，可用厘米(cm)或英寸(in)。高度和宽度也可用上述格式同时给出，这样可以改变原图的长宽比例。上述命令中的图片文件名是指欲插入的图片文件的文件名，图片必须是 eps 格式的。

(3) 用 epsfig 宏命令。

epsfig 宏命令的使用方法和 psfig 完全相同，具体方法如下。

首先需在 LaTeX 文档的文件说明部分加上：

```
\usepackage{epsfig}
```

然后在需要插入图片的地方引用：\epsfig{figure=图片文件名，height=高度} 或者

\epsfig{figure＝图片文件名,width＝宽度}。

其中的“高度”和“宽度”是指希望图片打印的高度和宽度,必须给出单位,可用厘米(cm)或英寸(in)。高度和宽度也可用上述格式给出,这样可以改变原图的长宽比例。上述命令中的图片文件名是指欲插入的图片文件的文件名,图片必须是 eps 格式的。

(4) 用 epsf 宏命令。

epsf 宏命令的使用方法是,首先需在 LaTeX 文档的文件说明部分加上:

```
\usepackage{epsf}
```

然后在需要插入图片的地方引用:\epsfxsize＝宽度/epsffile{图片文件名} 或者\epsfysize＝高度\epsffile{图片文件名}。

其中的“高度”和“宽度”是指希望图片打印的高度和宽度,必须给出单位,可用厘米(cm)或英寸(in)。高度和宽度也可用上述格式给出,这样可以改变原图的长宽比例。上述命令中的图片文件名是指欲插入的图片文件的文件名,图片必须是 eps 格式的。

3.3　子图的插入

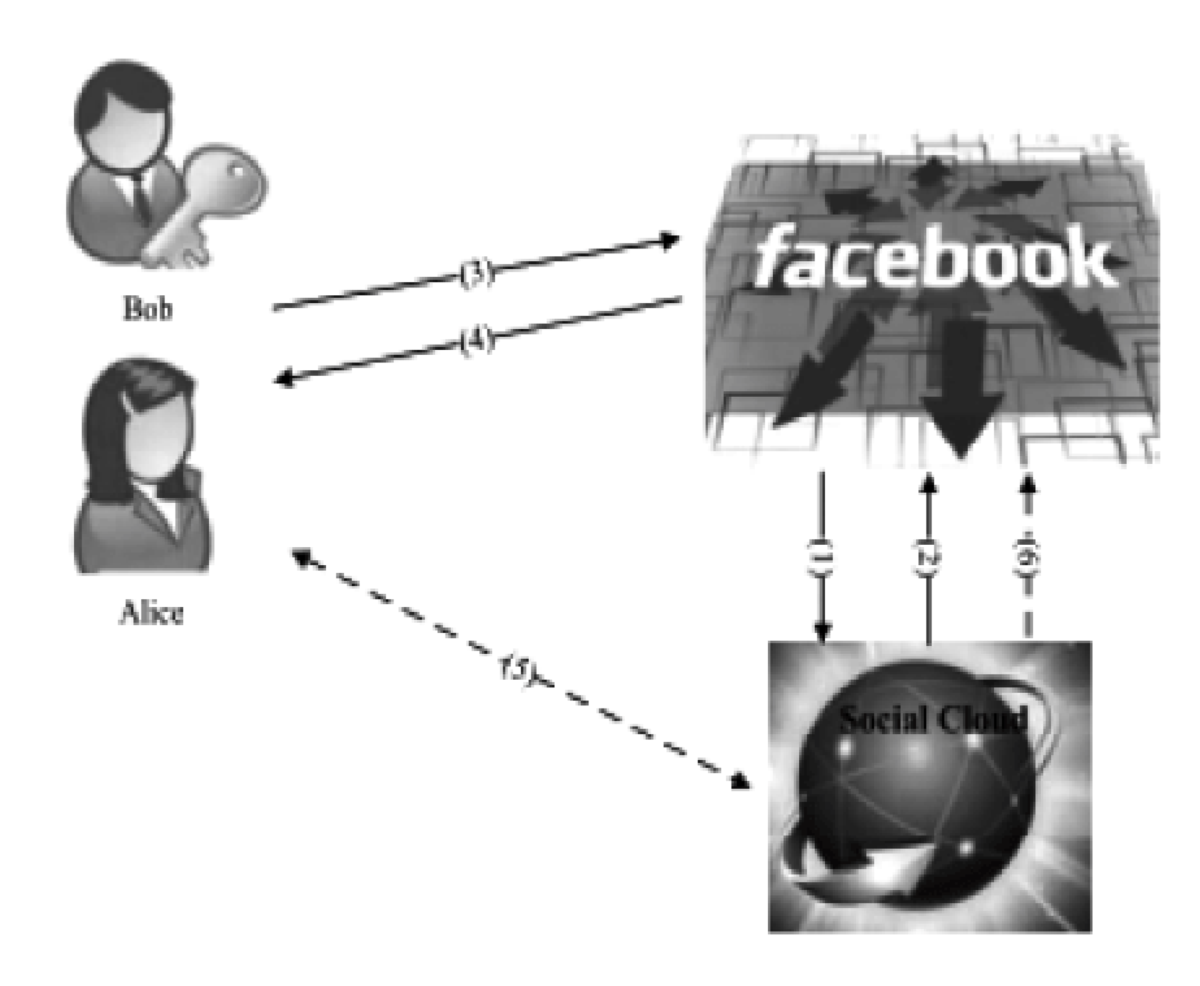
在科技文献中,有时为了对比图片,经常将几幅图片并列插入到文章中。这时需要用到 subfigure 宏包,在文件头部加上宏包,如图 3.15 所示。对应的命令行如图 3.16 所示。插入子图后的效果如图 3.17 所示。

```
\documentclass[10pt]{llncs}
\usepackage{llncsdoc}
\usepackage{amsmath,epsfig}
\usepackage{authblk}
\usepackage{amsfonts}
\usepackage{times}
\usepackage{bm}
\usepackage{slashbox}
\usepackage{subfigure}
```

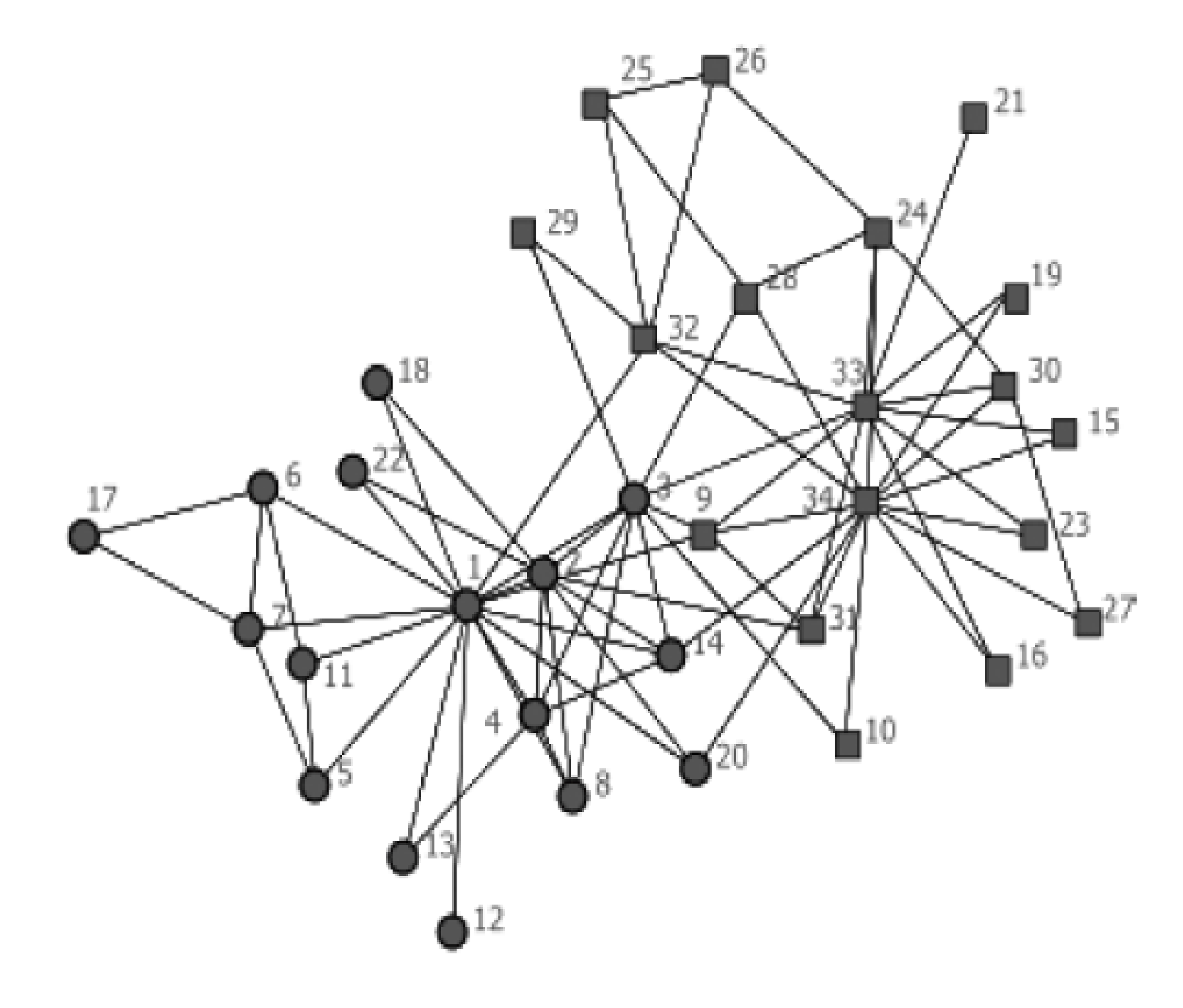
图 3.15　加入 subfigure 宏包

```
\begin{figure}[H]
\centering
\subfigure[SubfigureCaption]
{
\label{Fig.sub.1}
\includegraphics[scale=0.5]{architecture}
}
\subfigure[SubfigureCaption]
{
\label{Fig.sub.2}
\includegraphics[scale=0.5]{social}
}
\caption{MainfigureCaption}
\label{Fig.lable}
\end{figure}
```

图 3.16　插入子图命令行



(a) 子图一



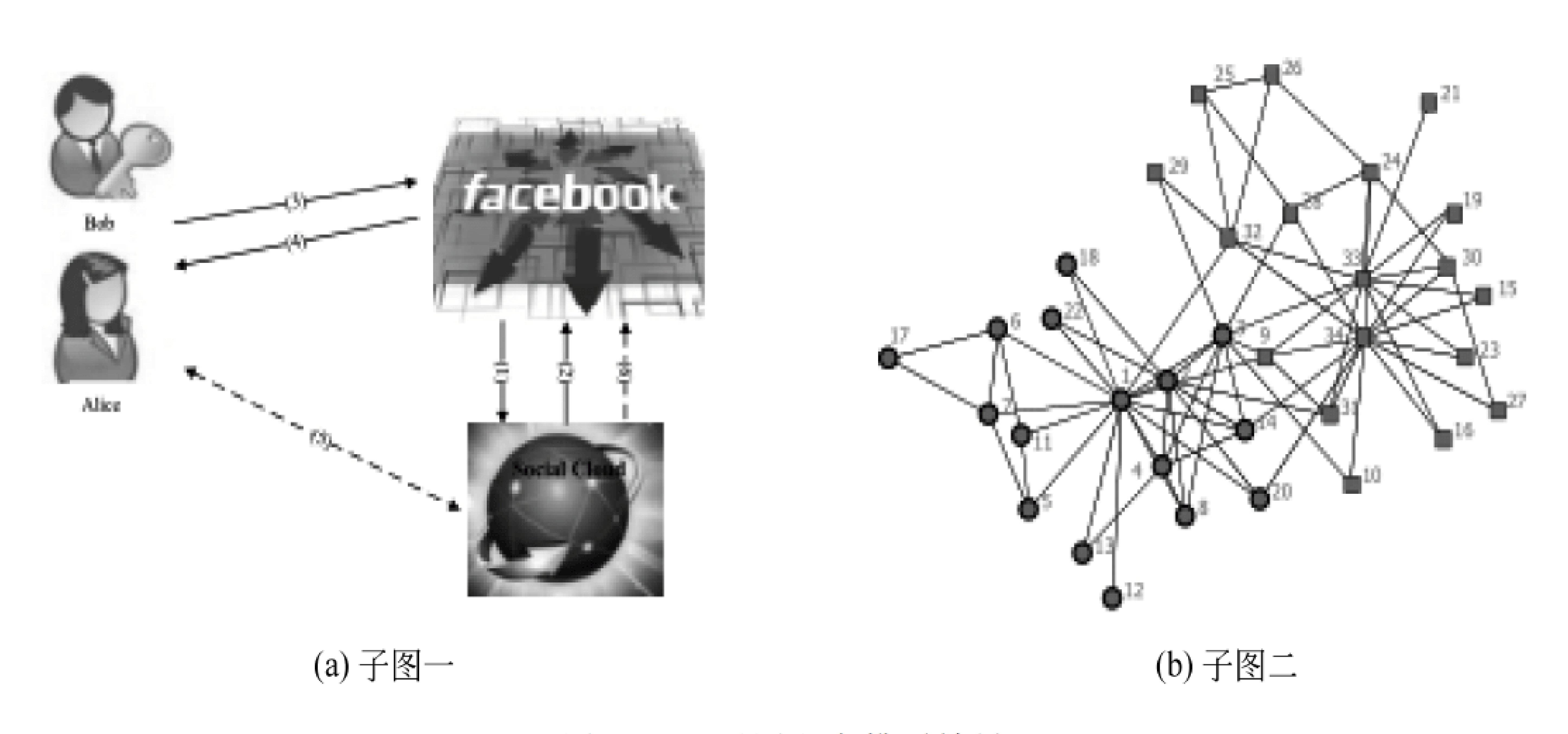
(b) 子图二

图 3.17　插入子图效果

图 3.17 中的两个子图是上下排列，如果希望两个子图是左右排列，则需要用到如图 3.18 所示的命令行。排列效果如图 3.19 所示。

```
\begin{figure}[htbp]
%\begin{minipage}{0.5\linewidth}
\centering
\subfigure[fig(a)]{
 \label{fig:subfig:a}
 \includegraphics[width=1.5in]{architecture}}
 %%\hspace{2in}
\subfigure[fig(b)]{
 \label{fig:subfig:b}
 \includegraphics[width=1.5in]{social}}
 %%\hspace{2in}
\caption{Original images }
 \label{fig:fig3}
\end{figure}
```

图 3.18　子图左右排列命令行



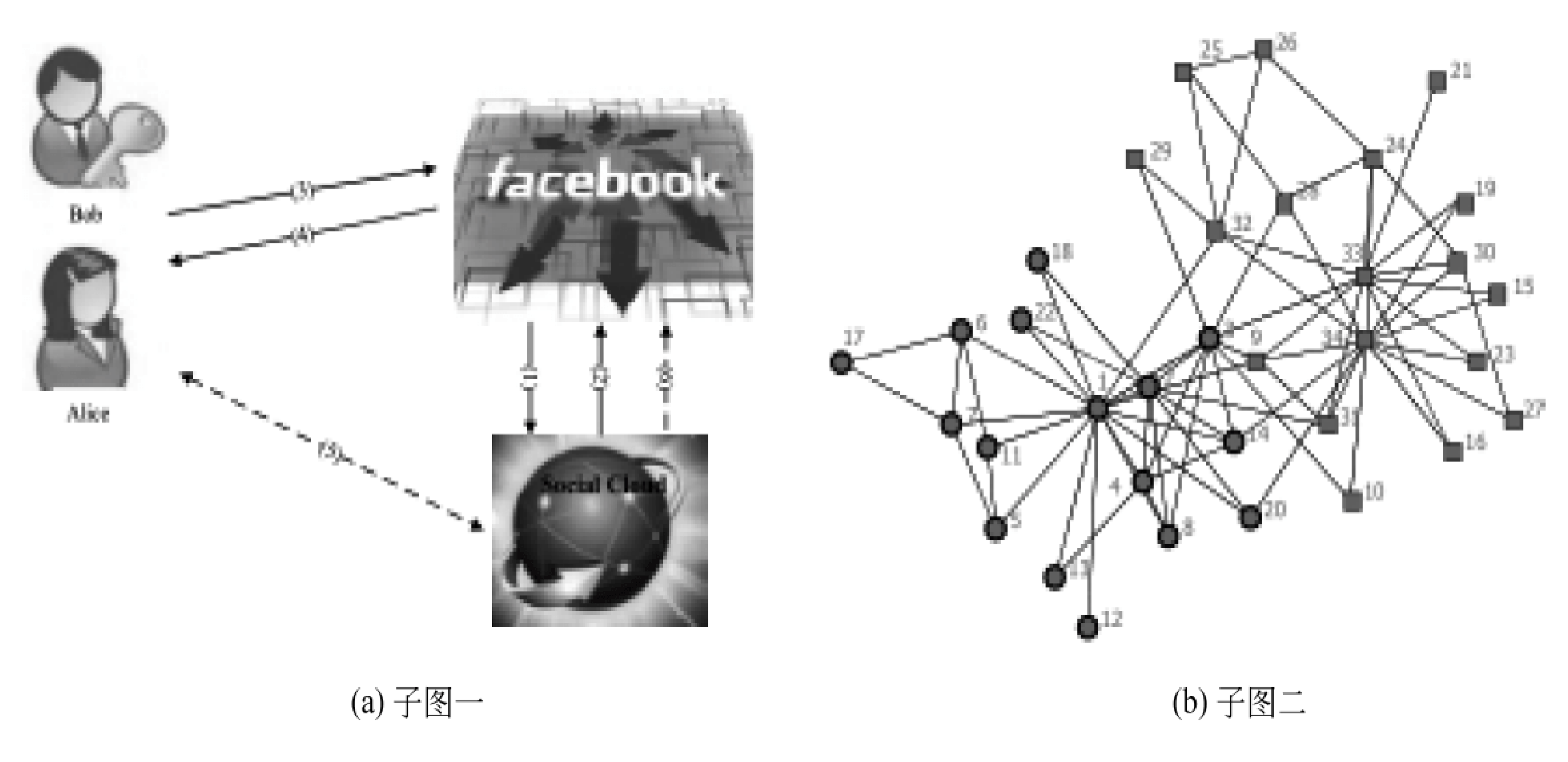
(a) 子图一　　　　(b) 子图二

图 3.19　子图左右排列效果

如果图片较多，需要多张照片排列，就需要用到如图 3.20 所示的命令行排版。排版效果如图 3.21 所示。

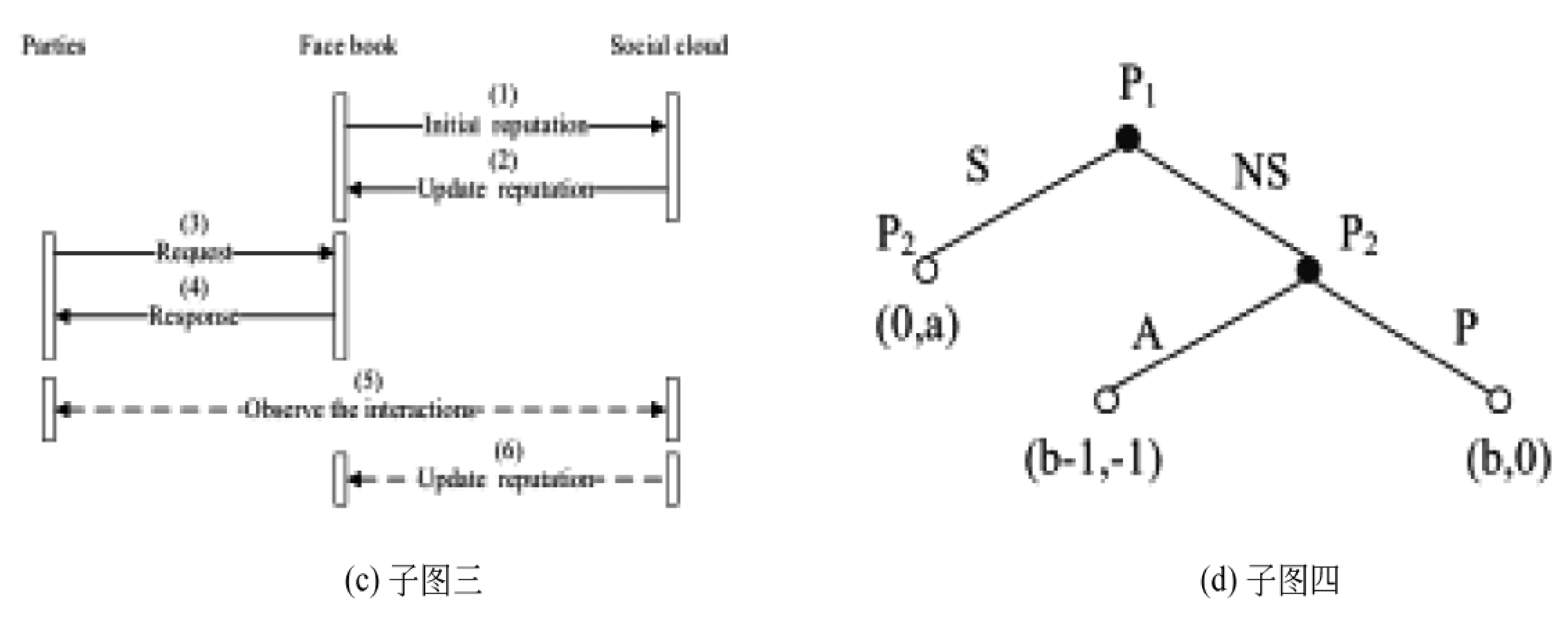
```
\begin{figure}[htbp]
\centering
\subfigure[fig(a)]{
 \label{fig:subfig:a}
 \includegraphics[width=1.5in]{architecture}}
\subfigure[fig(b)]{
 \label{fig:subfig:b}
 \includegraphics[width=1.5in]{social}}
\subfigure[fig(c)]{
 \label{fig:subfig:c}
 \includegraphics[width=1.5in]{interactions}}
\subfigure[fig(d)]{
 \label{fig:subfig:d}
 \includegraphics[width=1.5in]{wilful}}
\caption{Figure}
 \label{fig:fig4}
\end{figure}
```

图 3.20　多图排列命令行



(a) 子图一　　(b) 子图二

图 3.21　多图排列效果



(c) 子图三　　　　(d) 子图四

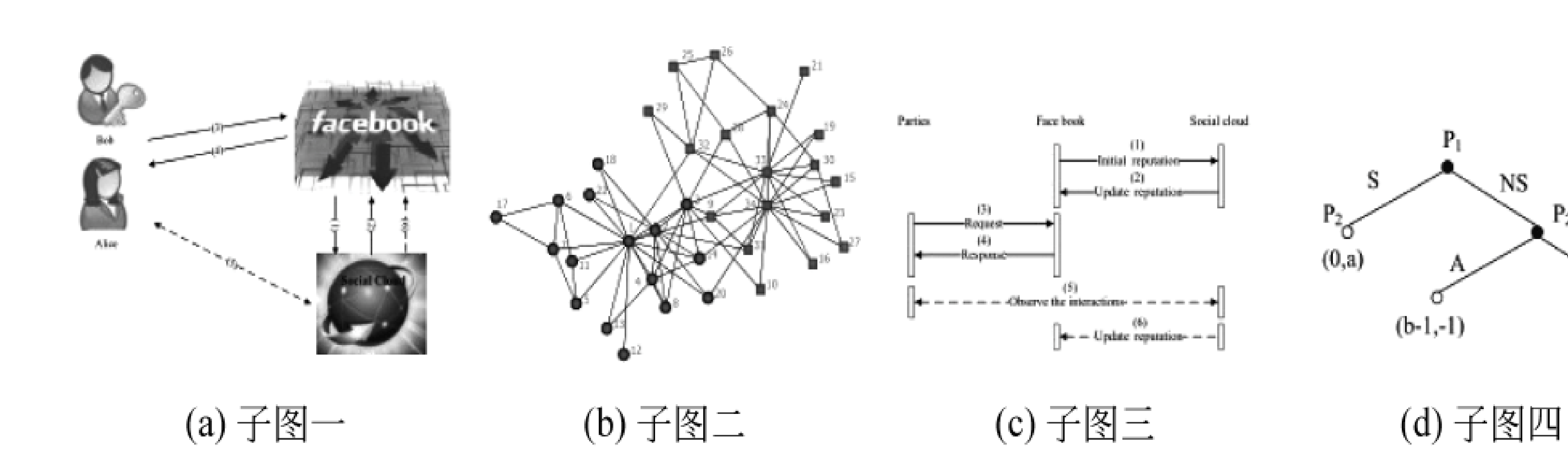
图 3.21（续）

3.4　图形的通栏问题

对于双栏的文章，如果图片太大，可以考虑将图片通栏放置，即一张图片占两栏，也称为通栏。对于这样的问题很容易解决，只需要在\begin{figure}和\end{figure}的 figure 后面加上 * 即可。对应命令行如图 3.22 所示，对应效果如图 3.23 所示。

```
\begin{figure*}[htbp]
\centering
\subfigure[fig(a)]{
 \label{fig:subfig:a}
 \includegraphics[width=1.5in]{architecture}}
\subfigure[fig(b)]{
 \label{fig:subfig:b}
 \includegraphics[width=1.5in]{social}}
\subfigure[fig(c)]{
 \label{fig:subfig:c}
 \includegraphics[width=1.5in]{interactions}}
\subfigure[fig(d)]{
 \label{fig:subfig:d}
 \includegraphics[width=1.5in]{wilful}}
\caption{Figure}
 \label{fig:fig4}
\end{figure*}
```

图 3.22　图形通栏命令行



(a) 子图一　(b) 子图二　(c) 子图三　(d) 子图四

model and propose an improved texture-preserved total variation (TPTV) image denoising model. Because control function of image structure is determined by image structure tensor, the diffusion process not only depends on the image gradient, but also on the local informa-

Ambient Intelligence and Humanized Computing, 5(5):621–622, 2014.

M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes,fluid dynamics, image and video inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–355. IEEE, 2001.

图 3.23　图形通栏效果

第 4 章　表格排版

4.1　表格基本命令

表格也是科技文献中常见的一种形式，它的排版和图形类似。插入表格的基本命令行如图 4.1 所示。

（1）\begin{table} …\end{table}表示插入一个表格。

（2）[!htp]用于设置表格的位置，详细说明参照 3.2.1 节。

（3）\centering 表示图片居中显示。

（4）\begin{tabular}…\ end{tabular}表示制表环境。LaTeX 提供了两种制表环境 array 和 tabular，它们的格式如下：

```
\begin{table}[!htp]
  \centering
  \begin{tabular}{lccc}
  \hline
    a1 & b1 &c1 & d1 \\ \hline
    a2 & b2 &c2 & d2 \\
    a3 & b3 &c3 & d3 \\
    a4 & b4 &c4 & d4 \\ \hline
  \end{tabular}
  \caption{title}
  \label{tab:example}
\end{table}
```

图 4.1　插入表格的基本命令行

```
\begin{array}[表格位置]{列样式} \end{array}
\begin{tabular}[表格位置]{列样式} \end{tabular}
```

这两个环境的选项和参数定义是相同的，区别在于 array 主要用于数组矩阵的排版，且只能用在数学环境中，如 equation 等。较为常用的是 tabular 制表环境。

（5）{lccc} 表示各列元素对齐方式，有 left-l、right-r、center-c 3 种取值。

（6）\hline 表示在此行下面画一横线。

（7）\\ 表示重新开始一行。

（8）& 表示列的分隔线。

（9）在\caption{title}中，title 表明表格的标题。

(10) \label{tab:example}中的 fig:example 表明表格在引用时的标示。

根据图 4.1 的命令行生成的表格如图 4.2 所示。

如果打算在每一行加一条横线，这在每一行后面添加\hline；如果打算给每一列加一条竖线，则在{lccc}每两个字母之间加符号|。例如，图 4.2 中如果给每一行每一列都加上横线和竖线，对应的命令行如图 4.3 所示，产生的表格效果如图 4.4 所示。

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

Table 1: title

图 4.2　表格示意图

```
\begin{table}[!htp]
    \centering
    \begin{tabular}{|l|c|c|c|}
     \hline
       a1   & b1   &c1  & d1 \\   \hline
       a2   & b2   &c2  & d2\\  \hline
       a3   & b3   &c3  & d3\\  \hline
       a4   & b4   &c4  & d4\\   \hline
    \end{tabular}
    \caption{title}
     \label{tab:example}
  \end{table}
```

图 4.3　表格对应的命令行(行和列均分隔开)

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

Table 1: title

图 4.4　表格效果图(行和列均分隔开)

4.2　表格字体的大小

表格默认的字体大小和正文字体相同，如果需要单独设置表格字体大小，需要在\begin{table}后加入控制字体大小的控制命令。图 4.5 给出了各种表格字体的示意图，用户可以根据自己的需要设置表格字体大小。

LaTeX 设置字体大小命令由小到大依次为

\tiny

\scriptsize

```
\footnotesize
\small
\normalsize
\large
\Large
\LARGE
\huge
\Huge
```

各种表格字体如图 4.5 所示。

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

Table 1: tiny

(a) tiny

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

Table 2: scriptsize

(b) scriptsize

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

Table 3: small

(c) small

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

Table 4: large

(d) large

图 4.5　各种表格字体

注意：其实字体的大小设置的命令行也适合正文字体的设置。

4.3　图形和表格在正文中的引用

图形和表格的格式设置好之后，最后一项工作是在正文中引用。在正文中引用一个图片的命令是\ref{fig:example}，引用一个表格的命令是\ref{tab:example}。LaTeX 与

Word 最大的不同是,前者对图片和表格自动编号,而且只要图片和表格的 label 不发生变化,即使正文中图片和表格的顺序出现了变化,或者图片和表格数目发生了变化,LaTeX 也会自动编号。而在 Word 中,如果图片和表格顺序或数目发生变化,需要用户自己调整图片和表格的编号。从这一点看来,LaTeX 确实很方便。

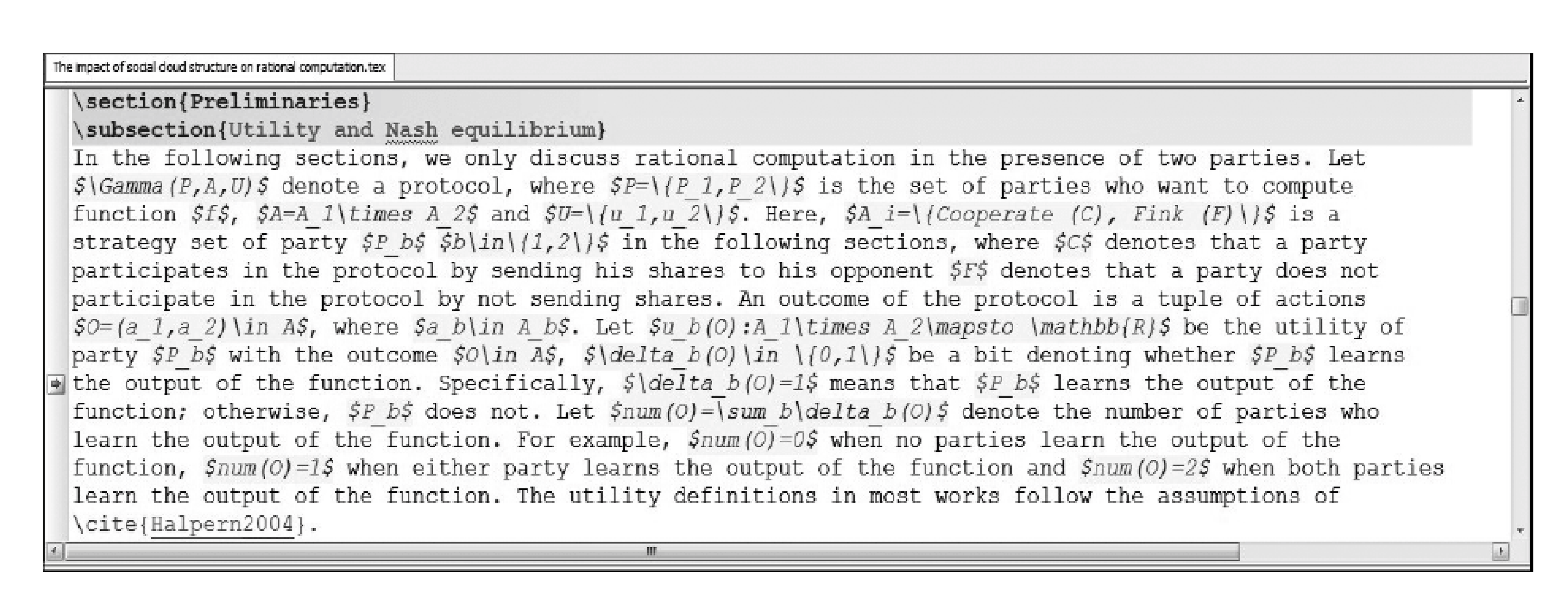
需要注意一点,表格也存在通栏的问题,只需要在\begin{table}和\end{table}中的 table 后面加上 * 即可。

第 5 章　数学公式与特殊符号

与 Word 相比，LaTeX 的一大优势就是数学公式的排版比较优美。在 Word 中，需要用户自己调节公式的大小和位置，在 LaTeX 中，这些都可以通过命令行的形式控制。

5.1　数学符号的基本显示

LaTeX 使用一种特殊的模式来排版数学符号和公式(mathematics)。段落中的数学表达式应该置于 \和\、$ 和 $ 或者\begin{math}和\end{math}之间。对于比较短的数学公式或者符号，使用 $ 和 $ 即可，如图 5.1 所示。对应 PDF 中的数学符号如图 5.2 所示。



```
The impact of social cloud structure on rational computation.tex
\section{Preliminaries}
\subsection{Utility and Nash equilibrium}
In the following sections, we only discuss rational computation in the presence of two parties. Let
$\Gamma(P,A,U)$ denote a protocol, where $P=\{P_1,P_2\}$ is the set of parties who want to compute
function $f$, $A=A_1\times A_2$ and $U=\{u_1,u_2\}$. Here, $A_i=\{Cooperate (C), Fink (F)\}$ is a
strategy set of party $P_b$ $b\in\{1,2\}$ in the following sections, where $C$ denotes that a party
participates in the protocol by sending his shares to his opponent $F$ denotes that a party does not
participate in the protocol by not sending shares. An outcome of the protocol is a tuple of actions
$O=(a_1,a_2)\in A$, where $a_b\in A_b$. Let $u_b(O):A_1\times A_2\mapsto \mathbb{R}$ be the utility of
party $P_b$ with the outcome $O\in A$, $\delta_b(O)\in \{0,1\}$ be a bit denoting whether $P_b$ learns
the output of the function. Specifically, $\delta_b(O)=1$ means that $P_b$ learns the output of the
function; otherwise, $P_b$ does not. Let $num(O)=\sum_b\delta_b(O)$ denote the number of parties who
learn the output of the function. For example, $num(O)=0$ when no parties learn the output of the
function, $num(O)=1$ when either party learns the output of the function and $num(O)=2$ when both parties
learn the output of the function. The utility definitions in most works follow the assumptions of
\cite{Halpern2004}.
```

图 5.1　简单数学符号命令行

图 5.1 中都是简单的数学符号，有些数学符号可以直接输入，例如 A、P 等，有些数学符号有特殊的命令对应，例如 Γ、δ 等，它们对应的命令为\Gamma、\delta。

在数学符号中常见的就是上标和下标。下标用符号“_”表示，上标用符号“^”表示。需要注意的是，如果下标符号不止一个，就需要将所有的下标用{}括起来。例如，A_{sum} 所对应的命令为 A_{sum}而不是 A_sum。如果输入 A_sum，所产生的数学符号为 A_sum。用户需要格外注意这个问题，因为出现这类错误，LaTeX 是不会报错的，因为 LaTeX 认

2.1 Utility and Nash equilibrium

In the following sections, we only discuss rational computation in the presence of two parties. Let $\Gamma(P, A, U)$ denote a protocol, where $P = \{P_1, P_2\}$ is the set of parties who want to compute function f, $A = A_1 \times A_2$ and $U = \{u_1, u_2\}$. Here, $A_i = \{Cooperate(C), Fink(F)\}$ is a strategy set of party P_b $b \in \{1, 2\}$ in the following sections, where C denotes that a party participates in the protocol by sending his shares to his opponent F denotes that a party does not participate in the protocol by not sending shares. An outcome of the protocol is a tuple of actions $O = (a_1, a_2) \in A$, where $a_b \in A_b$. Let $u_b(O) : A_1 \times A_2 \mapsto \mathbb{R}$ be the utility of party P_b with the outcome $O \in A$, $\delta_b(O) \in \{0, 1\}$ be a bit denoting whether P_b learns the output of the function. Specifically, $\delta_b(O) = 1$ means that P_b learns the output of the function; otherwise, P_b does not. Let $num(O) = \sum_b \delta_b(O)$ denote the number of parties who learn the output of the function. For example, $num(O) = 0$ when no parties learn the output of the function, $num(O) = 1$ when either party learns the output of the function and $num(O) = 2$ when both parties learn the output of the function. The utility definitions in most works follow the assumptions of [18].

图 5.2　简单数学符号效果

为这没有语法错误。

对于较大的数学式子，最好的方法是使用显示样式来排版：将它们放置于 \[和 \]、$ $ 和 $ $ 或\begin{displaymath}和\end{displaymath}之间，这样排版出的公式是没有编号的，如图 5.3 所示。图中的公式 $u_b(\sigma'_b, \sigma_{-b}) \leqslant u_b(\sigma)$没有编号。

Definition 1. *A strategy* $\boldsymbol{\sigma}$ *induces a* ***Nash equilibrium*** *if for every party* P_b *and any strategy* σ'_b*, it holds that*

$$u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) \leqslant u_b(\boldsymbol{\sigma}).$$

图 5.3　没有编号的公式

对于没有编号的公式，如果希望在后面引用该公式时，会带来麻烦。最好的情形是，给公式一个编号，这样在后面正文中如果需要再次用到该公式，可以直接提及该公式的编号。

如果希望公式有编号，可以使用 equation 环境来自动为每一个\begin{equation}和\end{equation}之间的公式编号。所对应的命令行如图 5.4 所示，产生的 PDF 文件中的公式如图 5.5 所示。图 5.5 与图 5.3 相比，只是公式 $u_b(\sigma'_b, \sigma_{-b}) \leqslant u_b(\sigma)$多了一个编号

(1)。在后面的正文中，如果希望再次引用该公式，就可以写“公式(1)”。

```
\textbf{Definition 1.} \emph{A strategy $\vec{\sigma}$ induces a \textbf{Nash equilibrium} if for every
party $P_b$ and any strategy $\sigma'_b$, it holds that
\begin{equation}
u_b(\sigma'_b,\vec{\sigma}_{-b})\leq u_b(\vec{\sigma}).
\end{equation}}
```

图 5.4　公式编号的命令行

Definition 1. *A strategy* $\boldsymbol{\sigma}$ *induces a* ***Nash equilibrium*** *if for every party* P_b *and any strategy* σ'_b*, it holds that*

$$u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) \leqslant u_b(\boldsymbol{\sigma}). \quad (1)$$

图 5.5　公式编号的效果图

在 LaTeX 命令行中，如果想在正文中引用带有编号的公式，不需要用户记住每个公式的详细编号，只需要知道每个公式的 label 即可。这一点与图形和表格的引用类似。例如，在图 5.6 中，公式 $u_b(\sigma'_b, \sigma_{-b}) \leqslant u_b(\sigma)$的 label 是 def1，用\label{def1}表示，在后面的文章中引用时，需要使用\ref{def1}引用，对应的命令行和所产生的效果如图 5.6 和 5.7 所示。

```
\textbf{Definition 1.} \emph{A strategy $\vec{\sigma}$ induces a
\textbf{Nash equilibrium} if for every party $P_b$ and any strategy $\sigma'_b$,
 it holds that
\begin{equation}
\label{def1}
u_b(\sigma'_b,\vec{\sigma}_{-b})\leq u_b(\vec{\sigma}).
\end{equation}}

In equation \ref{def1}, if $u_b(\sigma'_b,\vec{\sigma}_{-b})< u_b(\vec{\sigma})$ suffices, the strategy
$\vec{\sigma}$ induces a \textbf{ strict Nash equilibrium}. A (strict) Nash equilibrium guarantees that
no one gains any advantages by deviating from the equilibrium as long as other parties follow it. The
function of (strict) Nash equilibrium is to guarantee parties to follow the protocol.
```

图 5.6　公式标签的标示和引用

Definition 1. *A strategy* $\boldsymbol{\sigma}$ *induces a* ***Nash equilibrium*** *if for every party* P_b *and any strategy* σ'_b*, it holds that*

$$u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) \leqslant u_b(\boldsymbol{\sigma}). \quad (1)$$

In equation 1 if $u_b(\sigma'_b, \boldsymbol{\sigma}_{-b}) < u_b(\boldsymbol{\sigma})$ suffices, the strategy $\boldsymbol{\sigma}$ induces a **strict Nash equilibrium**. A (strict) Nash equilibrium guarantees that no one gains any advantages

图 5.7　公式引用后的效果

5.2 多行数学符号显示

对于一些复杂的数学公式，如果长度超过一行，可以考虑使用分行显示。可以使用 aligned 环境，将打算分行显示的公式输入在\begin{aligned}和\end{aligned}之间，如图 5.8 所示。在图 5.8 中，\\是换行符，& 是对齐符，想让公式在哪里对齐，就把 & 放到哪里。所产生的效果如图 5.9 所示。

```
\begin{displaymath}
 \begin{aligned}
&NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
&NU=\rho_1+\frac{\rho_3}{2} \\
&NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\\
&NU^{--}=-\rho_1+\rho_3
\end{aligned}
\end{displaymath}
```

图 5.8 使用 aligned 命令行

$$\begin{aligned}
&NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
&NU=\rho_1+\frac{\rho_3}{2} \\
&NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\\
&NU^{--}=-\rho_1+\rho_3
\end{aligned}$$

图 5.9 使用 aligned 分行后的效果

公式分行还可以使用 array 环境来排版数组(arrays)。它有些类似于 tabular 环境，使用\\命令来分行，如图 5.10 所示。

图 5.8 与图 5.10 的区别如下。

(1) 前者需要使用 & 符号来表示对齐位置，而后者不需要，因为后者类似于一个表格。注意，既然类似于表格，那么需要指定每一列的对齐方式，对齐方式的指定和表格相同，在图 5.10 中，使用 c，表明对齐方式是居中对齐，效果如图 5.11 所示。

```
\begin{displaymath}
 \begin{array}{c}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
NU=\rho_1+\frac{\rho_3}{2} \\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\\
NU^{--}=-\rho_1+\rho_3
\end{array}
\end{displaymath}
```

图 5.10 使用 array 命令行

$$\begin{array}{c}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
NU=\rho_1+\frac{\rho_3}{2} \\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\\
NU^{--}=-\rho_1+\rho_3
\end{array}$$

图 5.11 使用 array 分行后的效果

（2）前者无须指定列数，后者需要指定列数，并指定每一列的对齐方式。

使用 array 的另一个好处是对于数组或者矩阵的表示。例如，输入一个 3 行 3 列的矩阵，可以使用图 5.12 所示的命令行，通过命令行可以看出，这里每行每列数据的输入和表格类似。\left 和\right 分别表示左大括号和右大括号。生成 PDF 的效果如图 5.13 所示。

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
\rho_1&\rho_2&\frac{\rho_3}{3} \\
\rho_1&0&\frac{\rho_3}{2} \\
\rho_1&\rho_2&\frac{\rho_3}{2}\\
0&0&-\rho_1+\rho_3
\end{array} \right)
\end{displaymath}
```

图 5.12 矩阵对应命令行

$$X = \left(\begin{array}{ccc} \rho_1 & \rho_2 & \frac{\rho_3}{3} \\ \rho_1 & 0 & \frac{\rho_3}{2} \\ \rho_1 & \rho_2 & \frac{\rho_3}{2} \\ 0 & 0 & -\rho_1+\rho_3 \end{array} \right)$$

图 5.13 矩阵效果

像在 tabular 环境中一样，也可以在 array 环境中画线。例如，分隔矩阵中的元素命令行如图 5.14 所示，效果如图 5.15 所示。但是在数学公式中一般不习惯这样用。

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{c|c|c}
\rho_1&\rho_2&\frac{\rho_3}{3} \\
\hline
\rho_1&0&\frac{\rho_3}{2} \\
\hline
\rho_1&\rho_2&\frac{\rho_3}{2}\\
\hline
0&0&-\rho_1+\rho_3
\end{array} \right)
\end{displaymath}
```

图 5.14 矩阵分隔命令行

$$X = \left(\begin{array}{c|c|c} \rho_1 & \rho_2 & \frac{\rho_3}{3} \\ \hline \rho_1 & 0 & \frac{\rho_3}{2} \\ \hline \rho_1 & \rho_2 & \frac{\rho_3}{2} \\ \hline 0 & 0 & -\rho_1+\rho_3 \end{array} \right)$$

图 5.15 矩阵分隔效果

对于多行的公式也存在编号问题，情况相对来说比较复杂。对于分布于几行的公式或者方程组（equation system），可以使用 eqnarray 和 eqnarray * 环境来代替 equation。在 eqnarray 中，每一行都会有一个方程编号，在 eqnarray * 中不对方程进行编号。另外

使用\nonumber 命令也可以阻止 LaTeX 为公式生成一个编号。

eqnarray 和 eqnarray * 环境类似于 {rcl} 形式的三列表格。中间的一列可以用作等号或不等号，或者其他看起来适合的符号，使用 \\ 命令分行。图 5.16 给出了使用 eqnarray 多行公式编号命令行，图 5.17 是之后的运行效果。

```
\begin{eqnarray}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
NU=\rho_1+\frac{\rho_3}{2} \\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\\
NU^{--}=-\rho_1+\rho_3
\end{eqnarray}
```

图 5.16　使用 eqnarray 多行公式编号命令行

$$NU^+ = \rho_1 + \rho_2 + \frac{\rho_3}{3} \qquad (1)$$
$$NU = \rho_1 + \frac{\rho_3}{2} \qquad (2)$$
$$NU^- = -\rho_1 + \rho_2 + \frac{\rho_3}{2} \qquad (3)$$
$$NU^{--} = -\rho_1 + \rho_3 \qquad (4)$$

图 5.17　使用 eqnarray 多行公式编号效果

当多行公式都是独立公式时，这种编号方式比较合适，但是当多行公式同属一个公式，只是分行输入时，给每一行一个编号，显然不合适。因此需要为多行公式定义一个编号。

可以使用\nonumber，在前面几行的后面加上\nonumber，如图 5.18 所示，之后的效果如图 5.19 所示。

```
\begin{eqnarray}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3}\nonumber \\
NU=\rho_1+\frac{\rho_3}{2} \nonumber \\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2}\nonumber \\
NU^{--}=-\rho_1+\rho_3
\end{eqnarray}
```

图 5.18　使用 nonumber 为多行定义一个编号

$$\begin{aligned} NU^{+} &= \rho_1+\rho_2+\frac{\rho_3}{3} \\ NU &= \rho_1+\frac{\rho_3}{2} \\ NU^{-} &= -\rho_1+\rho_2+\frac{\rho_3}{2} \\ NU^{--} &= -\rho_1+\rho_3 \qquad (1) \end{aligned}$$

图 5.19 使用 nonumber 多行公式编号效果

但是图 5.19 的效果不好,用户希望将编号显示在公式的中间,可以使用 array 实现。命令行如图 5.20 所示,效果如图 5.21 所示。

```
\begin{eqnarray}
\begin{array}{l}
NU^+=\rho_1+\rho_2+\frac{\rho_3}{3} \\
NU=\rho_1+\frac{\rho_3}{2}\\
NU^-=-\rho_1+\rho_2+\frac{\rho_3}{2} \\
NU^{--}=-\rho_1+\rho_3
\end{array}
\end{eqnarray}
```

图 5.20 使用 array 为多方定义一个编号

$$\begin{array}{l} NU^{+}=\rho_1+\rho_2+\frac{\rho_3}{3} \\ NU=\rho_1+\frac{\rho_3}{2} \\ NU^{-}=-\rho_1+\rho_2+\frac{\rho_3}{2} \\ NU^{--}=-\rho_1+\rho_3 \end{array} \qquad (1)$$

图 5.21 使用 array 多行公式编号效果

在输入数学公式时,需要注意文字在数学环境和正文环境中的不同之处。例如在数学环境中:

(1) 空格和分行都将被忽略,也就是说,在数学环境下,LaTeX 不区分是否有空格,即使输入空格,LaTeX 也不识别。如果确实需要在数学环境中输入空格,则可以由特殊的命令来得到,例如\、\quad 或 \qquad。

(2) 不允许有空行,每个公式中只能有一个段落。

（3）每个字符都将被看作是一个变量名并以此来排版。如果希望在公式中出现普通的文本（使用正体字并可以有空格），那么用户必须使用命令\textrm{…} 来输入这些文本。命令行如图 5.22 所示，效果如图 5.23 所示。

```
\textbf{Definition 1.} \emph{A strategy $\vec{\sigma}$ induces a
\textbf{Nash equilibrium} if for every party $P_b$ and any strategy $\sigma'_b$,
 it holds that
\begin{equation}
\label{def1}
u_b(\sigma'_b,\vec{\sigma}_{-b})\leq u_b(\vec{\sigma}) \textrm{ for all } b\in \{0,1\}.
\end{equation}}
```

图 5.22　公式中文本正体显示命令行

Definition 1. *A strategy $\boldsymbol{\sigma}$ induces a* ***Nash equilibrium*** *if for every party P_b and any strategy σ'_b, it holds that*

$$u_b(\sigma'_b,\boldsymbol{\sigma}_{-b})\leqslant u_b(\boldsymbol{\sigma})\textit{for all}\, b\in\{0,1\}. \quad (1)$$

图 5.23　公式中文本正体显示效果图

一般的字符，在 amsmath 宏包中都可以找到，但是有些字体需要用到其他宏包。例如，表示实数集合的“空心粗体”（blackboard bold），这种字体可用 amsfonts 或 amssymb 宏包中的命令\mathbb 来得到。例如，将图 5.23 中的{0,1}变为空心 R，就可以使用图 5.24 所示的命令行，最后的效果如图 5.25 所示。

```
\textbf{Definition 1.} \emph{A strategy $\vec{\sigma}$ induces a
\textbf{Nash equilibrium} if for every party $P_b$ and any strategy $\sigma'_b$,
 it holds that
\begin{equation}
\label{def1}
u_b(\sigma'_b,\vec{\sigma}_{-b})\leq u_b(\vec{\sigma}) \textrm{ for all } b\in \mathbb{R}.
\end{equation}}
```

图 5.24　空心 R 的命令行

Definition 1. *A strategy $\boldsymbol{\sigma}$ induces a* ***Nash equilibrium*** *if for every party P_b and any strategy σ'_b, it holds that*

$$u_b(\sigma'_b,\boldsymbol{\sigma}_{-b})\leqslant u_b(\boldsymbol{\sigma})\textit{ for all}\, b\in\mathbb{R}. \quad (1)$$

图 5.25　空心 R 的效果

5.3 一些基本的数学符号

(1) 小写希腊字母(Lowercase Greek letters)的输入命令为 \alpha、\beta、\gamma…,大写希腊字母只要把命令的首字母大写即可,输入命令为 \Gamma、\Delta…

例如,希腊字母 λ、ξ、π、μ、Φ、Ω 所对应的命令如下：$ \lambda、\xi、\pi、\mu、\Phi、\Omega $ 。

(2) 平方根(square root)的输入命令为 \sqrt, n 次方根相应地为 \sqrt[n]。方根符号的大小由 LaTeX 自动加以调整,也可用 \surd 仅给出符号。

(3) 水平线命令 \overline 和 \underline 在表达式的上、下方画出水平线。

(4) 大括号命令 \overbrace 和 \underbrace 在表达式的上、下方给出一水平的大括号。

(5) 数学重音符号可覆盖多个字符的宽,重音符号可由 \widetilde 和 \widehat 等得到。

(6) 向量 (Vectors)通常用上方有小箭头(arrow symbols)的变量表示。这可由 \vec 得到。另两个命令 \overrightarrow 和 \overleftarrow 在定义从 A 到 B 的向量时非常有用。

一般情况下,乘法算式中的圆点符可以省略。然而有时为了帮助解读复杂的公式,也有必要用命令 \cdot 将圆点符表示出来。

(7) 函数名通常用罗马字体正体排版,而不是像变量名一样用意大利体排版。因此,LaTeX 提供下述命令来排版最重要的一些函数名。

```
\arccos \cos \csc \exp \ker \limsup \min
\arcsin \cosh \deg \gcd \lg \ln \Pr
\arctan \cot \det \hom \lim \log \sec
\arg \coth \dim \inf \liminf \max \sin
\sinh \sup \tan \tanh
```

(8) 排版模函数(modulo function)有两个命令：\bmod 用于二元运算符 $a \bmod b$,

\pmod 用于表达式，例如 $x \equiv a \pmod{b}$。

(9) 分数(fraction)使用 \frac{…}{…} 排版。一般来说，1/2 这种形式更受欢迎，因为对于少量的分式，它看起来更好些。

(10) 积分运算符 $\int$ (integral operator)用 \int 来生成。

(11) 求和运算符 $\sum$ (sum operator)由 \sum 生成。

(12) 乘积运算符×(product operator)由 \prod 生成。上限和下限用“ ^”和“_”来生成，类似于上标和下标。

(13) 对于括号(braces)和其他分隔符(delimiters)，在 TEX 中有各种各样的符号(例如 $[h\ k\ l]$)。圆括号和方括号可以用相应的键输入。花括号用 \{。其他的分隔符用专门命令(如 \updownarrow)来生成。

(14) 如果将命令 \left 放在分隔符前，TeX 会自动决定分隔符的正确大小。注意必须用对应的右分隔符 \right 来关闭每一个左分隔符 \left，并且只有当这两个分隔符排在同一行时大小才会被正确确定。

(15) 将 3 个圆点(three dots)输入公式可以使用几种命令。\ldots 将点排在基线上。\cdots 将它们设置为居中。除此之外，可用 \vdots 命令使其垂直，而用 \ddots 将得到对角型(diagonal dots)。

(16) 双引号。论文写作过程中可能会需要用到引号或者双引号，如果像 Word 一样直接敲入 " ，会出现错误。例如，如果在源文件中输入"emulates"，那么生成的 PDF 中会显示“emulates”。

在 LaTeX 中有专门的左引号和右引号。如果想在 PDF 中出现“emulates”的效果，应该在源文件中输入``emulates"。

需要注意的是：``不是两个单引号，而是键盘中 Tab 键上方的符号。

(17) 单引号在 LaTeX 中也不能直接用'emulates'，而是应该用`emulates'，对应的 PDF 中显示‘emulates’。注意 ` 是 Tab 键上方的符号，而'是键盘中的单引号键。

(18) 破折号和连字号。这些短划线的名称是：‘-’(连字号)、‘–’(短破折号)、‘——’(长破折号)和‘−’(减号)。

(19) 波浪号～。有时候网址中需要用到波浪号,但是在 LaTeX 源文件中直接输入～,编译后的 PDF 却不是这种结果,此时可以在～前面加一个符号\,变成\～。这一点与前面讲到的下划线类似。

(20) 度的符号(°)。在 LaTeX 中无法直接敲入该符号,可以使用\circ 表示度符号°。

(21) 省略号。在 LaTeX 中,省略号虽然可以直接通过键盘输入,但是前一个字母贴得非常紧,效果不好。LaTeX 有一个专门的命令输出省略号,即\ldots。

5.4　数学字体大小

在数学模式中,TeX 根据上下文选择字体大小。例如,使用较小的字体排版上标。如果想用罗马字体排版方程中的一部分,不要使用\textrm 命令,因为当\textrm 暂时脱离文本模式时字体大小交换机制不起作用。这时可以使用 \mathrm 来确保字体大小交换机制起作用。但是需要注意的是,\mathrm 只对于较短的项才起作用。空格仍然不起作用,并且重音字符也不起作用。

尽管如此,有时必须告诉 LaTeX 正确的字体大小。在数学模式中,字体大小用 4 个命令来设定:

```
\displaystyle (123), \textstyle (123), \scriptstyle (123) and
\scriptscriptstyle (123).
```

这一点类似于表格中控制字体大小。

5.5　定理、定义环境定义

在撰写科技论文时,除了一些数序符号和公式,可能还需要用到“引理”、“定义”、“公理”以及类似的结构。通常期刊或者会议提供的模板中没有给定这类结构的数学环境,因此需要用户自己定义。LaTeX 提供了\newtheorem 命令,通常该命令定义在\begin{document}之前,如图 5.26 所示。

\newtheorem 的一般格式如下:

```
\usepackage{float}
\usepackage{fancybox}
\usepackage{pdfpages}
\def\newblock{\hskip .11em plus .33em minus .07em}

\theoremstyle{TH}{
\newtheorem{lemma}{Lemma}
\newtheorem{theorem}[lemma]{Theorem}
\newtheorem{corrolary}[lemma]{Corrolary}
\newtheorem{conjecture}[lemma]{Conjecture}
\newtheorem{proposition}[lemma]{Proposition}
\newtheorem{claim}[lemma]{Claim}
\newtheorem{stheorem}[lemma]{Wrong Theorem}
\newtheorem{algorithm}{Algorithm}
}
\begin{document}
```

图 5.26　定义\newtheorem 环境

\newtheorem{*name*}[*counter*]{*text*}[*section*]

(1) *name* 是短关键字,用于标识"定理";*text* 定义"定理"的真实名称,会在最终文件中打印出来。例如,图 5.26 中\newtheorem{theorem}[lemma]{Theorem},在生成的 PDF 中,会显示 Theorem 而不是 theorem。

(2) 方括号中的选项是任意的,可以用于指定"定理"中使用的标号。*counter* 可以指定先前声明的"定理"的 *name*。然后新"定理"会按同样的顺序编号;*section* 指定"定理"编号所在的章节层次。

在文件的导言中执行 \newtheorem 命令之后,在文件中可以使用如下命令。例如,图 5.27 和 5.29 分别给出了引理和定理的命令行,图 5.28 和图 5.30 给出了命令行对应的效果。需要注意的是,引理、定理以及定义等也是自动编号。如果在后面的正文中需要引用该引理、定理或定义,也可以使用\ref{label}引用,前提条件是在命令行中要给出它们的 label,这一点类似于图形、表格和公式的引用。

针对图 5.29 和图 5.30,需要注意以下几点。

(1) 在图 5.29 中使用的是\begin{theorem}而不是\begin{Theorem},这是因为图 5.26 中的 *name* 项是 theorem。

(2) 在图 5.30 中显示的是 Theorem 而不是 theorem,是因为图 5.26 中的 *text* 项是

```
\begin{lemma}
Given $m^*=1+(2a-4c+2\gamma)/\gamma$, where $m^*$ denotes the remain iterations in second stage of the
protocol, there exists a sequential equilibrium such that both parties cooperate before $t^*=n-m^*$
iterations, where $n$ denotes the total iterations of the second stage.
\end{lemma}
```

图 5.27　引理环境命令行

Lemma 1: *Given* $m^* = 1 + (2a - 4c + 2\gamma)/\gamma$, *where* m^* *denotes the remain iterations in second stage of the protocol, there exists a sequential equilibrium such that both parties cooperate before* $t^* = n - m^*$ *iterations, where* n *denotes the total iterations of the second stage.*

图 5.28　引理环境效果

```
\begin{theorem}
Given the utility assumptions such as correctness and exclusivity, there exists a fairly rational
computation protocol in the hybrid world with $n>1+(2a-4c+2\gamma)/\gamma$ constant iterations under
incomplete information both in fail-stop setting and Byzantine setting, where a party is a TFTer party
with probability $\gamma$. If enhanced trapdoor permutations exist, the fairly rational computation
protocol also establish in the real world.
\end{theorem}
```

图 5.29　定理环境命令行

Theorem 2: *Given the utility assumptions such as correctness and exclusivity, there exists a fairly rational computation protocol in the hybrid world with* $n > 1 + (2a - 4c + 2\gamma)/\gamma$ *constant iterations under incomplete information both in fail-stop setting and Byzantine setting, where a party is a TFTer party with probability* γ*. If enhanced trapdoor permutations exist, the fairly rational computation protocol also establish in the real world.*

图 5.30　定理环境效果

Theorem。

(3) 在原文中,只有一个引理和一个定理,但是图 5.30 中的定理编号为 2,也就是说它是接续引理 1 而来。这是因为在图 5.26 中,\newtheorem{theorem}[lemma]{Theorem}的 counter 项为[lemma],这意味着定理的编号是接续引理的。因此在文章中引理 1 后面就是定理 2。如果想让定理单独编号,只需将\newtheorem{theorem}[lemma]{Theorem}改为\newtheorem{theorem}{Theorem}即可,这意味着定理重新编号。

(4) 如果希望定理的编号以章节为单位,可以将图 5.26 中的\newtheorem{theorem}{Theorem} 改为\newtheorem{theorem}{Theorem}[section]。运行后的效果如图 5.31 所示。

4.3 Analysis of the protocol

Theorem 4.1: *Given the utility assumptions such as correctness and exclusivity, there exists a fairly rational computation protocol in the hybrid world with* $n > 1 + (2a - 4c + 2\gamma)/\gamma$ *constant iterations under incomplete information both in fail-stop setting and Byzantine setting, where a party is a TFTer party with probability* γ*. If enhanced trapdoor permutations exist, the fairly rational computation protocol also establish in the real world.*

图 5.31 定理以章节为单位编号

(5) 以上几点说明同样适用与引理、定义以及其他自定义的数学环境。

5.6 数学符号表

图 5.32～图 5.50 中将给出在数学模式中常用的所有符号。使用这些图中列出的符号,必须事先安装 AMS 数学字库并且在文档的导言区加载宏包 amssymb。如果系统中没有安装 AMS 宏包和数学字库,可去下述地址下载:CTAN:/tex-archive/macros/latex/required/amslatex。

$\hat{a}$	\hat{a}	$\check{a}$	\check{a}	$\tilde{a}$	\tilde{a}	$\acute{a}$	\acute{a}
$\grave{a}$	\grave{a}	$\dot{a}$	\dot{a}	$\ddot{a}$	\ddot{a}	$\breve{a}$	\breve{a}
$\bar{a}$	\bar{a}	$\vec{a}$	\vec{a}	$\widehat{A}$	\widehat{A}	$\widetilde{A}$	\widetilde{A}

图 5.32 数学模式重音符

α	\alpha	θ	\theta	o	o	υ	\upsilon
β	\beta	ϑ	\vartheta	π	\pi	ϕ	\phi
γ	\gamma	ι	\iota	ϖ	\varpi	φ	\varphi
δ	\delta	κ	\kappa	ρ	\rho	χ	\chi
ϵ	\epsilon	λ	\lambda	ϱ	\varrho	ψ	\psi
ε	\varepsilon	μ	\mu	σ	\sigma	ω	\omega
ζ	\zeta	ν	\nu	ς	\varsigma		
η	\eta	ξ	\xi	τ	\tau		

图 5.33 小写希腊字母

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

图 5.34　大写希腊字母

$<$	`<`	$>$	`>`	$=$	`=`
$\leq$	`\leq` or `\le`	$\geq$	`\geq` or `\ge`	$\equiv$	`\equiv`
$\ll$	`\ll`	$\gg$	`\gg`	$\doteq$	`\doteq`
$\prec$	`\prec`	$\succ$	`\succ`	$\sim$	`\sim`
$\preceq$	`\preceq`	$\succeq$	`\succeq`	$\simeq$	`\simeq`
$\subset$	`\subset`	$\supset$	`\supset`	$\approx$	`\approx`
$\subseteq$	`\subseteq`	$\supseteq$	`\supseteq`	$\cong$	`\cong`
$\sqsubset$	`\sqsubset` [a]	$\sqsupset$	`\sqsupset` [a]	$\Join$	`\Join` [a]
$\sqsubseteq$	`\sqsubseteq`	$\sqsupseteq$	`\sqsupseteq`	$\bowtie$	`\bowtie`
$\in$	`\in`	$\ni$	`\ni` , `\owns`	$\propto$	`\propto`
$\vdash$	`\vdash`	$\dashv$	`\dashv`	$\models$	`\models`
$\mid$	`\mid`	$\parallel$	`\parallel`	$\perp$	`\perp`
$\smile$	`\smile`	$\frown$	`\frown`	$\asymp$	`\asymp`
$:$	`:`	$\notin$	`\notin`	$\neq$	`\neq` or `\ne`

[a]：使用宏包 latexsym 来得到这个符号。

图 5.35　二元关系符

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$\ast$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

图 5.36　二元运算符

$\sum$	`\sum`	$\bigcup$	`\bigcup`	$\bigvee$	`\bigvee`	$\bigoplus$	`\bigoplus`
$\prod$	`\prod`	$\bigcap$	`\bigcap`	$\bigwedge$	`\bigwedge`	$\bigotimes$	`\bigotimes`
$\coprod$	`\coprod`	$\bigsqcup$	`\bigsqcup`			$\bigodot$	`\bigodot`
$\int$	`\int`	$\oint$	`\oint`			$\biguplus$	`\biguplus`

图 5.37　大尺寸运算符

$\leftarrow$	\leftarrow or \gets	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\rightarrow$	\rightarrow or \to	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons	$\iff$	\iff (bigger spaces)	$\leadsto$	\leadsto [a]

[a]: 使用宏包 latexsym 来得到这个符号。

图 5.38 箭头

<table>
<tr><td>(</td><td>(</td><td>)</td><td>)</td><td>↑</td><td>\uparrow</td><td>⇑</td><td>\Uparrow</td></tr>
<tr><td>[</td><td>[or \lbrack</td><td>]</td><td>] or \rbrack</td><td>↓</td><td>\downarrow</td><td>⇓</td><td>\Downarrow</td></tr>
<tr><td>{</td><td>\{ or \lbrace</td><td>}</td><td>\} or \rbrace</td><td>↕</td><td>\updownarrow</td><td>⇕</td><td>\Updownarrow</td></tr>
<tr><td>⟨</td><td>\langle</td><td>⟩</td><td>\rangle</td><td>|</td><td>| or \vert</td><td>‖</td><td>\| or \Vert</td></tr>
<tr><td>⌊</td><td>\lfloor</td><td>⌋</td><td>\rfloor</td><td>⌈</td><td>\lceil</td><td>⌉</td><td>\rceil</td></tr>
<tr><td>/</td><td>/</td><td>\</td><td>\backslash</td><td></td><td>. (dual. empty)</td><td></td><td></td></tr>
</table>

图 5.39 定界符

$\lgroup$	\lgroup	$\rgroup$	\rgroup	$\lmoustache$	\lmoustache	$\rmoustache$	\rmoustache
$\arrowvert$	\arrowvert	$\Arrowvert$	\Arrowvert	$\bracevert$	\bracevert		

图 5.40 大尺寸定界符

$\dots$	\dots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\hbar$	\hbar	$\imath$	\imath	$\jmath$	\jmath	ℓ	\ell
$\Re$	\Re	$\Im$	\Im	$\aleph$	\aleph	$\wp$	\wp
$\forall$	\forall	$\exists$	\exists	$\mho$	\mho [a]	∂	\partial
$'$	'	$\prime$	\prime	$\emptyset$	\emptyset	∞	\infty
∇	\nabla	$\triangle$	\triangle	$\Box$	\Box [a]	$\Diamond$	\Diamond [a]
$\bot$	\bot	$\top$	\top	$\angle$	\angle	$\surd$	\surd
$\diamondsuit$	\diamondsuit	$\heartsuit$	\heartsuit	$\clubsuit$	\clubsuit	$\spadesuit$	\spadesuit
$\neg$	\neg or \lnot	$\flat$	\flat	$\natural$	\natural	$\sharp$	\sharp

[a]: 使用宏包 latexsym 来得到这个符号。

图 5.41 其他符号

这些符号也可以在文本模式中使用。

$\dag$	\dag	$\S$	\S	©	\copyright
$\ddag$	\ddag	$\P$	\P	£	\pounds

图 5.42　非数学符号

$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner	$\lrcorner$	\lrcorner
$\lvert$	\lvert	$\rvert$	\rvert	$\lVert$	\lVert	$\rVert$	\rVert

图 5.43　定界符

$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth	$\daleth$	\daleth	$\gimel$	\gimel

图 5.44　AMS 希腊和希伯来字母

$\dashleftarrow$	\dashleftarrow	$\dashrightarrow$	\dashrightarrow	$\multimap$	\multimap
$\leftleftarrows$	\leftleftarrows	$\rightrightarrows$	\rightrightarrows	$\upuparrows$	\upuparrows
$\leftrightarrows$	\leftrightarrows	$\rightleftarrows$	\rightleftarrows	$\downdownarrows$	\downdownarrows
$\Lleftarrow$	\Lleftarrow	$\Rrightarrow$	\Rrightarrow	$\upharpoonleft$	\upharpoonleft
$\twoheadleftarrow$	\twoheadleftarrow	$\twoheadrightarrow$	\twoheadrightarrow	$\upharpoonright$	\upharpoonright
$\leftarrowtail$	\leftarrowtail	$\rightarrowtail$	\rightarrowtail	$\downharpoonleft$	\downharpoonleft
$\leftrightharpoons$	\leftrightharpoons	$\rightleftharpoons$	\rightleftharpoons	$\downharpoonright$	\downharpoonright
$\Lsh$	\Lsh	$\Rsh$	\Rsh	$\rightsquigarrow$	\rightsquigarrow
$\looparrowleft$	\looparrowleft	$\looparrowright$	\looparrowright	$\leftrightsquigarrow$	\leftrightsquigarrow
$\curvearrowleft$	\curvearrowleft	$\curvearrowright$	\curvearrowright		
$\circlearrowleft$	\circlearrowleft	$\circlearrowright$	\circlearrowright		

图 5.45　AMS 箭头

$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\doteqdot$	\doteqdot or \Doteq
$\leqslant$	\leqslant	$\geqslant$	\geqslant	$\risingdotseq$	\risingdotseq
$\eqslantless$	\eqslantless	$\eqslantgtr$	\eqslantgtr	$\fallingdotseq$	\fallingdotseq
$\leqq$	\leqq	$\geqq$	\geqq	$\eqcirc$	\eqcirc
$\lll$	\lll or \llless	$\ggg$	\ggg or \gggtr	$\circeq$	\circeq
$\lesssim$	\lesssim	$\gtrsim$	\gtrsim	$\triangleq$	\triangleq
$\lessapprox$	\lessapprox	$\gtrapprox$	\gtrapprox	$\bumpeq$	\bumpeq
$\lessgtr$	\lessgtr	$\gtrless$	\gtrless	$\Bumpeq$	\Bumpeq
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\thicksim$	\thicksim
$\lesseqqgtr$	\lesseqqgtr	$\gtreqqless$	\gtreqqless	$\thickapprox$	\thickapprox
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\approxeq$	\approxeq
$\curlyeqprec$	\curlyeqprec	$\curlyeqsucc$	\curlyeqsucc	$\backsim$	\backsim
$\precsim$	\precsim	$\succsim$	\succsim	$\backsimeq$	\backsimeq
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Vdash$	\Vdash
$\Subset$	\Subset	$\Supset$	\Supset	$\Vvdash$	\Vvdash
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\backepsilon$	\backepsilon
$\therefore$	\therefore	$\because$	\because	$\varpropto$	\varpropto
$\shortmid$	\shortmid	$\shortparallel$	\shortparallel	$\between$	\between
$\smallsmile$	\smallsmile	$\smallfrown$	\smallfrown	$\pitchfork$	\pitchfork
$\vartriangleleft$	\vartriangleleft	$\vartriangleright$	\vartriangleright	$\blacktriangleleft$	\blacktriangleleft
$\trianglelefteq$	\trianglelefteq	$\trianglerighteq$	\trianglerighteq	$\blacktriangleright$	\blacktriangleright

图 5.46　AMS 二元关系符

$\nless$	\nless	$\ngtr$	\ngtr	$\varsubsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\varsupsetneqq$	\varsupsetneqq
$\nleq$	\nleq	$\ngeq$	\ngeq	$\nsubseteqq$	\nsubseteqq
$\nleqslant$	\nleqslant	$\ngeqslant$	\ngeqslant	$\nsupseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\nleqq$	\nleqq	$\ngeqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\nsim$	\nsim
$\nprec$	\nprec	$\nsucc$	\nsucc	$\ncong$	\ncong
$\npreceq$	\npreceq	$\nsucceq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\ntriangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\ntriangleright$	\ntriangleright
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\ntrianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\ntrianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

表 5.47 AMS 二元否定关系符和箭头

$\dotplus$	\dotplus	$\centerdot$	\centerdot	$\intercal$	\intercal
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\Cup or \doublecup	$\Cap$	\Cap or \doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes	$\circledast$	\circledast
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge		

图 5.48 AMS 二元运算符

$\hbar$	\hbar	$\hslash$	\hslash	$\Bbbk$	\Bbbk
$\square$	\square	$\blacksquare$	\blacksquare	$\circledS$	\circledS
$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle	$\complement$	\complement
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\Game$	\Game
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\angle$	\angle	$\measuredangle$	\measuredangle	$\sphericalangle$	\sphericalangle
$\diagup$	\diagup	$\diagdown$	\diagdown	$\backprime$	\backprime
$\nexists$	\nexists	$\Finv$	\Finv	$\varnothing$	\varnothing
$\eth$	\eth	$\mho$	\mho		

图 5.49 AMS 其他符号

例子	命令	所需宏包
ABCdef	`\mathrm{ABCdef}`	
ABCdef	`\mathit{ABCdef}`	
$ABCdef$	`\mathnormal{ABCdef}`	
$\mathcal{ABC}$	`\mathcal{ABC}`	
$\mathscr{ABC}$	`\mathcal{ABC}`	mathrsfs
$\mathcal{ABC}$	`\mathcal{ABC}`	eucal with option: mathcal or
	`\mathscr{ABC}`	eucal with option: mathscr
$\mathfrak{ABCdef}$	`\mathfrak{ABCdef}`	eufrak
$\mathbb{ABC}$	`\mathbb{ABC}`	amsfonts or amssymb

图 5.50　数学字母

第6章 参考文献

参考文献是科技论文中必不可少的一部分，参考文献的引用从一个方面反映出作者的严谨态度。一般来说，在论文中凡是引用他人已发表甚至未发表文献中的概念、定义以及定理等资料，都需要在引用处明确标示，并且在论文尾部列出所引用文献列表。

科技论文中通常在以下几个地方引用参考文献。

(1) 在作者的研究基础中，会引用作者已有的一些相关工作，反映研究工作的继承性，或者引用之前作者的研究成果，引出不足或待改进之处，表明该研究是对以前研究的一种改进和延伸。

(2) 在相关研究成果中，一般会引用他人已有的研究成果，主要论述之前工作的主要结论以及需要改进之处。如果只引用他人的结论而不列出参考文献，会被认为是剽窃行为。

(3) 对于科技论文中出现的一些概念以及定义，有时因为篇幅所限，不方便一一列出，此时可以引用相关文献，便于读者有据可查。如果读者希望进一步了解相关的定义，可以查阅相关文献。

在 Word 中，也可以通过交叉引用的方式自动生成参考文献，但是，一旦参考文献的编号出现变动，在 Word 中更新参考文献标号时，会出现找不到对应编号的情况。此时，必须对参考文献重新编号，重新进行交叉引用。而在 LaTeX 中就没有这种问题，只要 LaTeX 所引用的参考文献标示不改变，LaTeX 自动对参考文献编号，无须作者手工调整。在 LaTeX 中，常用的参考文献引用方式有两种。

6.1　LaTeX 默认环境

此时,LaTeX 模板中参考文献编写的命令如下:

```
\begin{thebibliography}{编号样本}
\bibitem[记号]{引用标志 1}文献条目 1
\bibitem[记号]{引用标志 2}文献条目 2
⋮
\end{thebibliography}
```

在这个格式中,每一项的说明如下。

编号样本:表示参考文献项目符号的形式,例如,{99}表示参考文献的项目是按照数字进行编号,并且最多允许列出 99 个参考文献,如果需要列出更多的参考文献,可以修改这个数字。

引用标志:可以唯一标示本条文献,引用标志不能重复。在正文中引用该文献时,使用\cite{引用标志}可以引用该文献。

文献条目:包括标题、作者、期刊、年代和页码。

默认参考文献的行间距为一行,有时候显得太大了。可以在\begin{thebibliography}{} 后添加\addtolength{\itemsep}{-1.5ex}来缩小行间距。-1.5ex 表示每行缩小 1.5ex。

注意:

(1) 在引用参考文献时\begin{thebibliography}和\end{thebibliography}一定要成对出现,否则会出现错误。

(2) thebibliography 其实是一个枚举环境,因此对于 itemize 和 enumerate,可以用同样的方法缩小行间距。

例 6.1　一个 thebibliography 下的参考文献实例。

```
\begin{thebibliography}{99}
\bibitem{r1}
\textit{Scientific Style and Format: The CBE manual for authors,
editors and publishers}. Style Manual Committee, Council of Biology Editors.
```

```
Sixth ed. Cambridge University Press, 1994.
\bibitem{r2}
L.U. Ante, Cem surgere: Surgite postquam sederitis, qui manducatis panem doloris,
\textit{Omnes} \textbf{13} (1916), 114--119.
\bibitem{r3}
T.X. Confortavit, \textit{Seras}, Portarum, New York, 1995.
\bibitem{r4}
P.A. Deus, Ater hoc et filius et mater praestet nobis,
\textit{Paterhoc} \textbf{66} (1993), 856--890.
\end{thebibliography}
```

从图 6.1 可以看出，在所列文献中，斜体和黑体的地方需要作者利用\textit{}和\textbf{}修饰一下。另外，每一条文献的内容也需要作者一一录入，这样大大增加了参考文献的复杂程度。如果论文中引用的参考文献较多，可以使用 EndNote，它是用于处理大批量文献管理的一种工具。EndNote 适用于大量文献的处理，尤其当有文献稍做改动时，其优点更明显。

References

[1] *Scientific Style and Format: The CBE manual for authors, editors and publishers*. Style Manual Committee, Council of Biology Editors. Sixth ed. Cambridge University Press, 1994.

[2] L.U. Ante, Cem surgere: Surgite postquam sederitis, qui manducatis panem doloris, *Omnes* **13** (1916), 114–119.

[3] T.X. Confortavit, *Seras*, Portarum, New York, 1995.

[4] P.A. Deus, Ater hoc et filius et mater praestet nobis, *Paterhoc* **66** (1993), 856–890.

图 6.1 生成参考文献示例

6.2 参考文献管理工具 Bibtex

期刊的 LaTeX 模板中参考文献编写的命令是\bibliography{bib 文件名}，其中，bib 表示参考文献所在的文件；bst 表示参考文献样式文件。

一般情况下 bst 由系统提供，所以不需要编写，不过当发表期刊时，期刊一般会提供样式文件给你，毕竟各个期刊对参考文献的要求不一样。例如，使用 IEEE 的模板，读者

可以从网站上下载 IEEE_CS_Latex8.5x11x2.zip,在这个压缩包里有一个 IEEEtran.bst 文件,在这个文件里,可以设置 IEEE 模板中参考文献的格式,如图 6.2 所示。其中下划线部分表示期刊的卷号、期号和页码表示的形式。用此模板在正文中生成的参考文献格式如图 6.3 所示。可以看出期刊的期卷号使用 no.和 vol.表示,页码用 pp 表示。

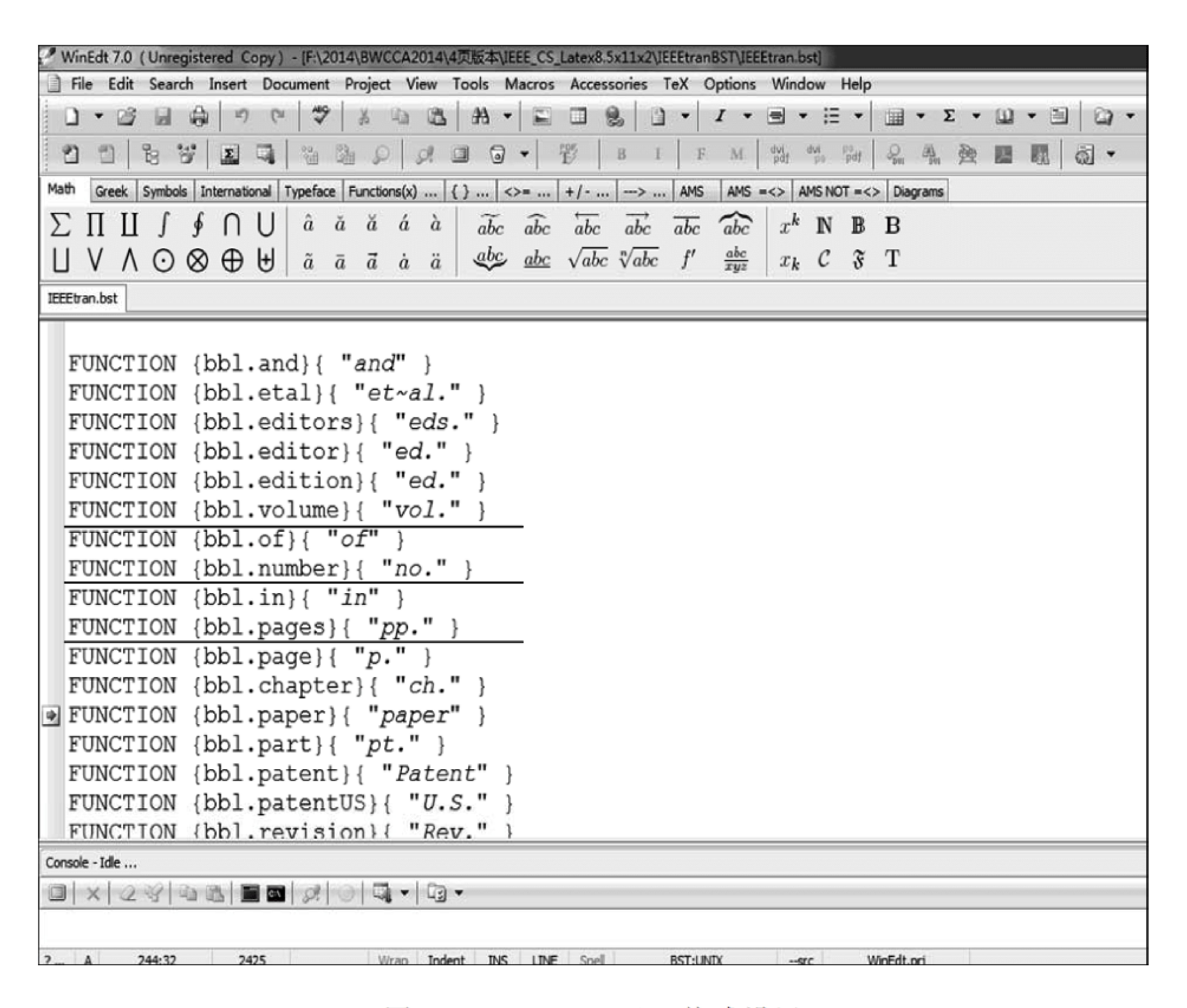


图 6.2　IEEEtran.bst 格式设置

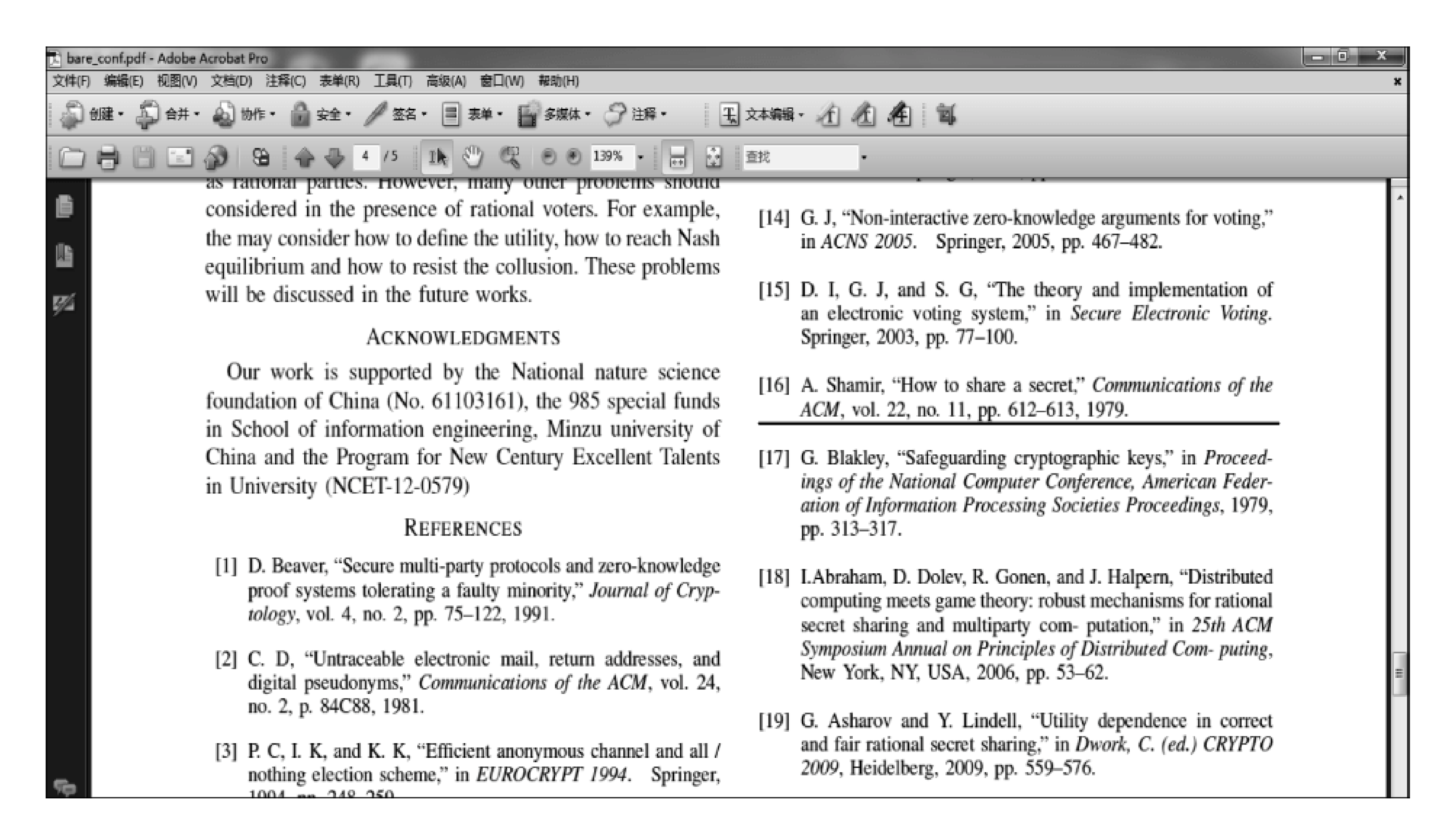
注意:如果需要改变期卷号或页码的显示方式,可以修改 IEEEtran.bst 文件的对应位置,但是一般不建议读者修改 bst 文件。

为了使用第二种方法自动生成参考文献,除了在文件夹中包含 bst 文件以外,还需要其他的工作。

(1) 在文件的头部需要包含宏包 natbib。具体操作如下:在\begin{document}之前加入语句\usepackage{natbib}。

(2) IEEEtran.bst 表示参考文献的类型,如果在文章中使用该模板,需要在文件头部

标识。文献类型的位置如图 6.4 所示。



considered in the presence of rational voters. For example, the may consider how to define the utility, how to reach Nash equilibrium and how to resist the collusion. These problems will be discussed in the future works.

ACKNOWLEDGMENTS

Our work is supported by the National nature science foundation of China (No. 61103161), the 985 special funds in School of information engineering, Minzu university of China and the Program for New Century Excellent Talents in University (NCET-12-0579)

REFERENCES

[1] D. Beaver, "Secure multi-party protocols and zero-knowledge proof systems tolerating a faulty minority," *Journal of Cryptology*, vol. 4, no. 2, pp. 75–122, 1991.

[2] C. D, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, p. 84C88, 1981.

[3] P. C, I. K, and K. K, "Efficient anonymous channel and all / nothing election scheme," in *EUROCRYPT 1994*. Springer,

[14] G. J, "Non-interactive zero-knowledge arguments for voting," in *ACNS 2005*. Springer, 2005, pp. 467–482.

[15] D. I, G. J, and S. G, "The theory and implementation of an electronic voting system," in *Secure Electronic Voting*. Springer, 2003, pp. 77–100.

[16] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[17] G. Blakley, "Safeguarding cryptographic keys," in *Proceedings of the National Computer Conference, American Federation of Information Processing Societies Proceedings*, 1979, pp. 313–317.

[18] I.Abraham, D. Dolev, R. Gonen, and J. Halpern, "Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty com- putation," in *25th ACM Symposium Annual on Principles of Distributed Com- puting*, New York, NY, USA, 2006, pp. 53–62.

[19] G. Asharov and Y. Lindell, "Utility dependence in correct and fair rational secret sharing," in *Dwork, C. (ed.) CRYPTO 2009*, Heidelberg, 2009, pp. 559–576.

图 6.3　IEEEtran.bst 对应的参考文献格式

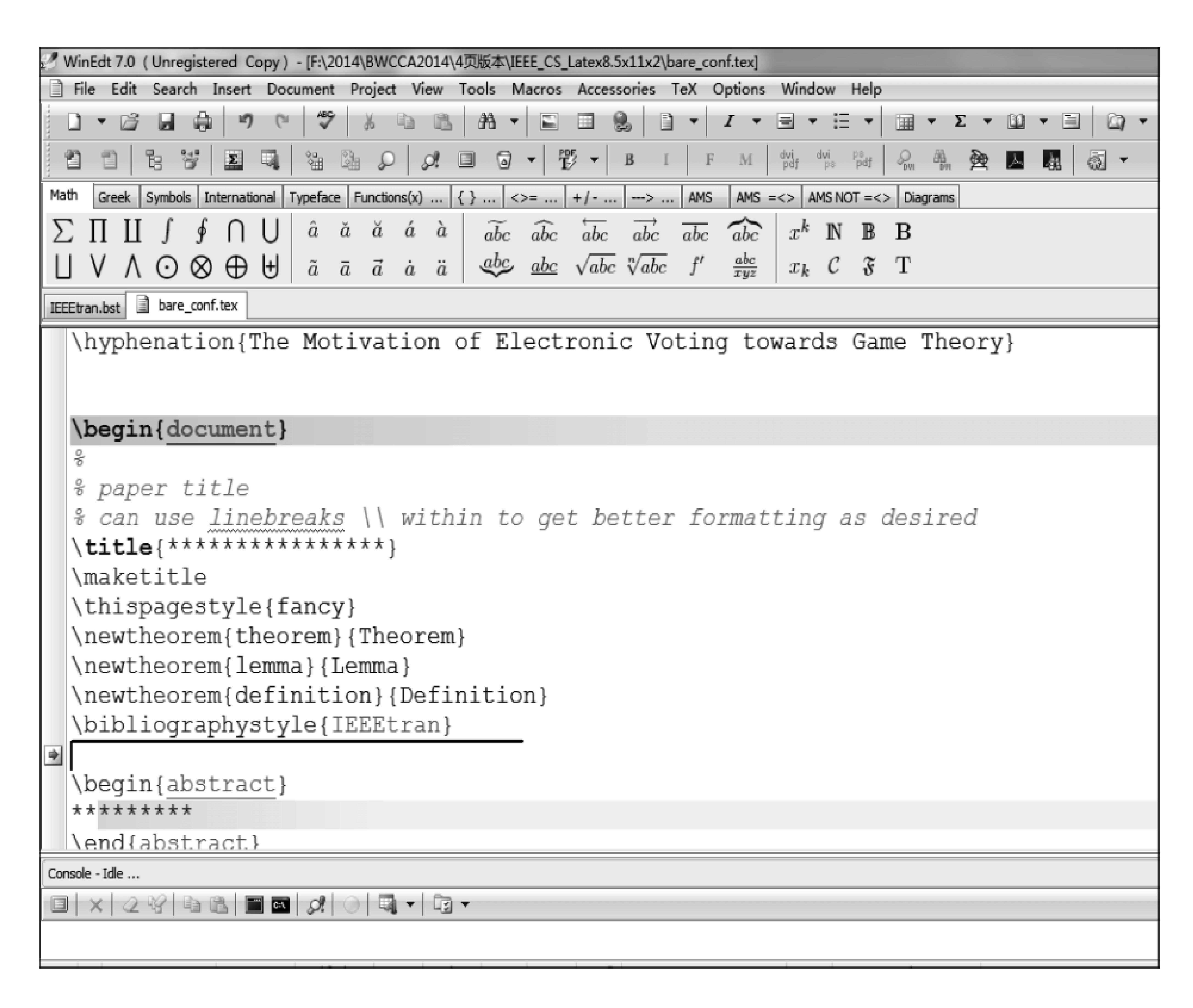


图 6.4　文献类型的位置

如果期刊没有提供 bst 文件,可以使用 LaTeX 默认的 bst 文件,标准类型为 plain。引用语句为

```
\bibliographystyle{plain}
```

除了标准类型外,还有其他的类型。

① unsrt:基本上跟 plain 类型一样,除了参考文献的条目的编号是按照引用的顺序,而不是按照作者的字母顺序。

② alpha:类似于 plain 类型,当参考文献的条目的编号基于作者名字和出版年份的顺序。

③ abbrv:缩写格式。

(3) 标记引用(make citations)。当在文档中想使用引用时,插入 LaTeX 命令\cite{标号}。

例如,\cite{Gettys90},根据 cls 文件的不同定制,以及在正文引用格式的不同要求,也可能是 citet 命令或者 citep 命令。

"标记"就是前边定义在@article 后面的名称。

(4) 告诉 LaTeX 生成参考文献列表。在 LaTeX 的结束前输入如下命令行:

```
\bibliographystyle{jmb}
\bibliography{./MYREFS}
```

文件中应包含 jmb. bst(定义参考文献类型)文件和 MYREFS. bib 文件。

运行 BibTeX 分为下面四步。

(1) 用 LaTeX 编译. tex 文件,这是生成一个. aux 的文件,这告诉 BibTeX 将使用哪些引用。

(2) 用 BibTeX 编译. bib 文件。

(3) 再次用 LaTeX 编译. tex 文件,这时在文档中已经包含了参考文献,但此时引用的编号可能不正确。

(4) 最后用 LaTeX 编译. tex 文件,如果一切顺利的话,这时所有东西都已正常了。

BibTeX 文件的后缀名为 . bib 。先来看一个例子:

```
@article{Gettys90,
author= {Jim Gettys and Phil Karlton and Scott McGregor},
title= {The {X} Window System, Version 11},
journal= {Software Practice and Experience},
volume= {20},
number= {S2},
year= {1990},
abstract= {A technical overview of the X11 functionality. This is an update of the X10 TOG paper by
Scheifler \& Gettys.}
}
```

说明：

第一行@article 告诉 BibTeX 这是一个文章类型的参考文献。还有其他格式，例如 article、book、booklet、conference、inbook、incollection、inproceedings、manual、misc、mastersthesis、phdthesis、proceedings、techreport、unpublished 等接下来的 Gettys90，就是在正文中引用这个条目的名称。其他就是参考文献里面的具体内容。

为了在 LaTeX 中使用 BibTeX 数据库，必须先做下面三件事情。

(1) 设置参考文献的类型(bibliography style)。标准的为 plain：

```
\bibliographystyle{plain}
```

其他的类型包括 unsrt，unsrt 基本上跟 plain 类型一样，除了参考文献的条目的编号是按照引用的顺序排列，而不是按照作者的字母顺序排列。

alpha 类似于 plain 类型，当参考文献的条目的编号基于作者名字和出版年份的顺序. abbrv-缩写格式。

(2) 标记引用 (make citations)。当在文档中想使用引用时，插入 LaTeX 命令\cite{引用文章名称}，“引用文章名称”就是前边定义在@article 后面的名称。

(3) 告诉 LaTeX 生成参考文献列表。在 LaTeX 的结束前输入：

```
\bibliography{bibfile}
```

这里 bibfile 就是 BibTeX 数据库文件 bibfile. bib。

例子：将上面的 BibTeX 的例子保存为 bibfile. bib。

```
\documentclass{article}
    \begin{document}
        We cite \cite{name1} and \cite{name2}
     \bibliography{bibfile}
    \bibliographystyle{plain}
    \end{document}
```

将上面的内容保存为 bibtex-example. tex。LaTeX 编译一次，BibTeX 编译一次，再用 LaTeX 编译两次就大功告成了！

获取 Bib 文件使用如下网址（可以将 Springer 文献格式变成 BibTeX）：

http://www. it. usyd. edu. au/～niu/cgi-bin/springer. cgi。

关于文献类型：

@article 条目为期刊或杂志上的一篇文章。

不可少域 author、title、journal、year。

可省略域 volume、number、pages、month、note。

```
@article{name,
author={a and b},
title={title},
journal={journal name},
volume={42},
number={1},
year={2008},
issn={0110-0101},
pages={1--8},
doi={http://doi.xxx.org},
publisher={ACM},
address={New York, NY, USA},
};
```

@book 条目为有确定出版社的书籍。

不可少域 author 或 editor、title、publisher、year。

可省略域 volume 或 number、series、address、edition、month、note。

@booklet 条目为印制的有封皮的作品，但没有出版社或赞助机构的名称。

不可少域 title。

可省略域 author、howpublished、address、month、year、note。

@conference 与下面的@inproceedings 相同。

@inbook 条目为一本书的一部分(章、节或某些页)。

不可少域 author 或 editor、title、chapter 或 pages、publisher、year。

可省略域 volume 或 number、series、type、address、edition、month、note。

@incollection 条目为一本书中有自己题目的一部分。

不可少域 author、title、booktitle、publisher、year。

可省略域 editor、volume 或 number、series、type、chapter、pages、address、edition、month、note。

@inproceedings 条目为会议论文集中的一篇文章。

不可少域 author、title、booktitle、year。

可省略域 editor、volume 或 number、series、pages、address、month、organization、publisher、note。

```
@InProceedings{b07name,
  author={a and b},
      title={title},
      booktitle={Proceedings of the conference},
      address={Sydney, Australia},
      month=Nov,
      year=2008,
      pages={1-2},
      affiliation={Uni-name, Country},
      URL={http://url/}
  };
```

@manual 条目为科技文档。

不可少域 title。

可省略域 author、organization、address、edition、month、year、note。

@mastersthesis 条目为硕士论文。

不可少域 author、title、school、year。

可省略域 type、address、month、note。

@misc 条目为不属于其他任何类型的作品。

不可少域没有。

可省略域 author、title、howpublished、month、year、note。

```
@misc{name,
    author="a",
    title="title",
    howpublished="Website",
    year={2008},
    note={\url{https://www.mysite.org}}
};
```

@phdthesis 条目为博士论文。

不可少域 author、title、school、year。

可省略域 type、address、month、note。

@proceedings 条目为会议论文集。

不可少域 title、year。

可省略域 editor、volume 或 number、series、address、month、organization、publisher、note。

@techreport 条目为学校或其他研究机构印制的报告。

不可少域 author、title、institution、year。

可省略域 type、number、address、month、note。

```
@techreport{name,
author={a and b}
title={title},
institution={institution rpt no.}
year={2008}
};
```

@unpublished 条目为有作者和标题的还未出版的作品。

不可少域 author、title、note。

可省略域 month、year。

在每项条目中还可以有可省略域 key 和 crossref。

6.3 文献样式

文献样式由 bst 文件控制。一般有 plain、abbrv、alpha、unsrt 等，BibTeX Style Examples 给出很多常见样式的例子。要制定自己的样式文件也很容易，需要 makebst 程序。

如果自己输入每一个文献，是非常烦琐的事情，而且格式未必符合要求，这里介绍一个简单易行的方法。

(1) 首先打开 Google 学术页面，如图 6.5 所示，如果 Google 学术登录不了，可以选择谷粉学术页面，如图 6.6 所示。



图 6.5 Google 学术页面



图 6.6 谷粉学术页面

(2) 输入文章的题目，按 Enter 键。

(3) 搜索结果如图 6.7 所示。



图 6.7 搜索结果

(4) 单击“引用”,会在新弹出窗口中显示引用格式。如果在 Word 中引用这些文献,可以直接按照模板要求复制过去,如果使用 LaTeX 模板里的 BibTeX,可以直接单击图 6.8 中的 BibTeX。

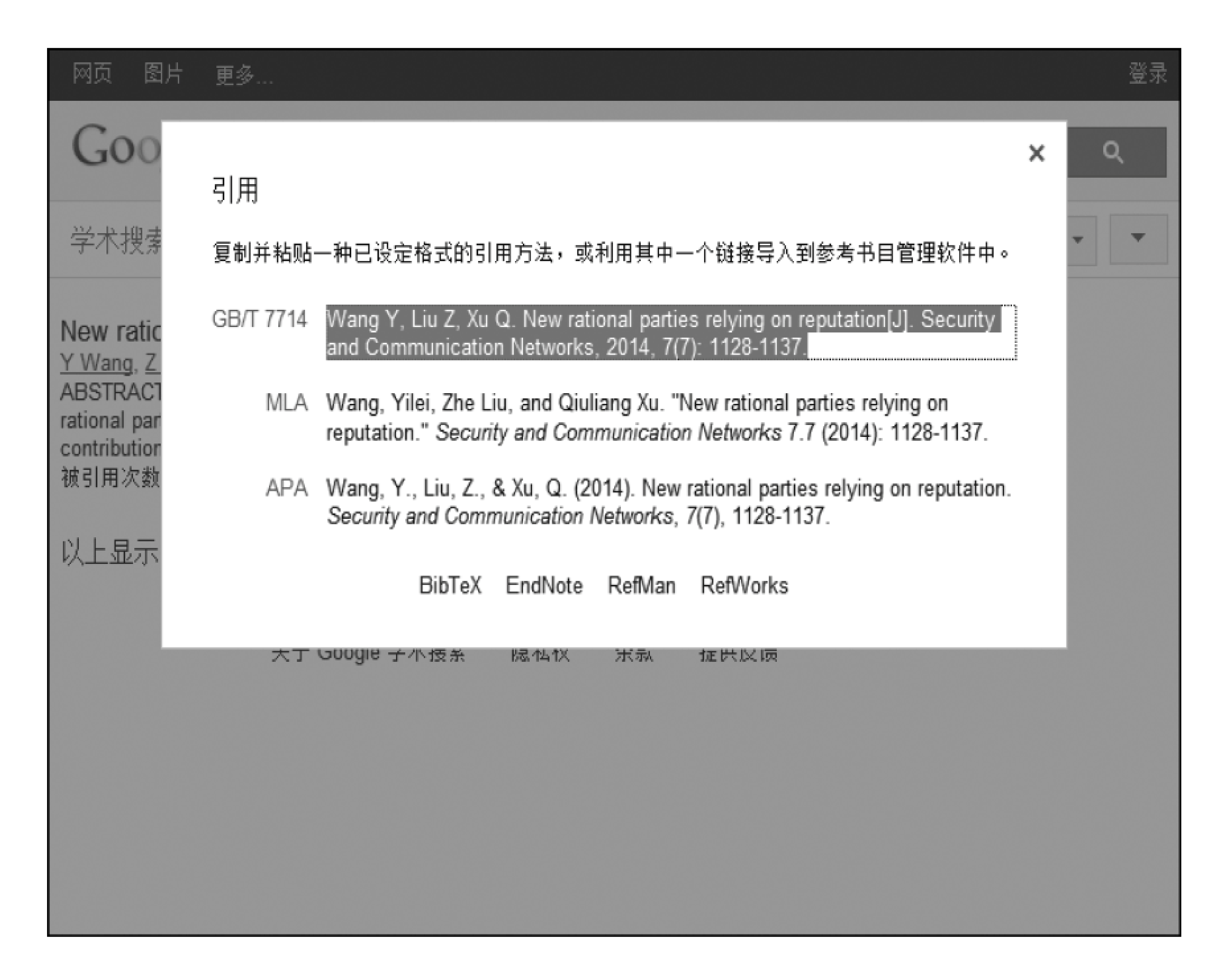


图 6.8 引用情况

(5) 将弹出页面中的文字(见图 6.9)复制到 Bib 文件中即可。

```
@article{wang2014new,
  title={New rational parties relying on reputation},
  author={Wang, Yilei and Liu, Zhe and Xu, Qiuliang},
  journal={Security and Communication Networks},
  volume={7},
  number={7},
  pages={1128--1137},
  year={2014},
  publisher={Wiley Online Library}
}
```

图 6.9 文献对应的引用命令

建立好 Bib 文件之后,接下来的工作是在正文中引用文献。引用文献的基本命令是 \cite{文献标示},在图 6.9 中文献标示是 wang2014new。在文献中的相应位置填入\cite{wang2014new},如图 6.10 所示。

```
  parties in most time. However, sometimes
  they also have incentives to cheat as
  malicious parties.
\end{itemize}

Recently, rational parties are proposed as new
kinds of parties in secure computing protocols
\cite{Halpern2010,Halpern2004,wang2014new,wang2015fair,
Whenever they adopt an action in the computing
protocol, they have corresponding utilities. Note
that one basic goal of rational parties is to
maximize their utilities. To achieve this goal,
rational parties may dynamically change their
strategies according to other parties' reactions.
```

图 6.10　参考文献对应命令

参考文献在正文中的效果如图 6.11 所示。

Recently, rational parties are proposed as new kinds of parties in secure computing protocols Halpern and Pass (2010); Halpern and Teague (2004); Wang et al. (2014, 2015a); Chen et al. (2012b). Whenever they adopt an action in the cpmputing protocol, they have corresponding utilities. Note that one basic goal of rational parties is to maximize their utilities. To achieve this goal, rational parties may dynamically change their strategies according to other parties' reactions. More

图 6.11　参考文献正文引用效果

参考文献部分的形式如图 6.12 所示。需要注意的是，不同的模板，参考文献的格式不同，通常不需要自己更改参考文献的格式，如果需要重新设置格式，需要更改.bst 文件。

有时，模板中还会使用\citet{文献标示}和\citep{文献标示}引用参考文献。\citet{文献标示}和\citep{文献标示}的细微区别如表 6.1 所示。

U. Straccia. Towards a fuzzy description logic for the semantic web (preliminary report). In *The Semantic Web: Research and Applications*, pages 167–181. Springer, 2005.

V. Vychodil T. Kuhr. Fuzzy logic programming reduced to reasoning with attribute implications. *Fuzzy Sets and Systems*, 262:1–20, 2015.

A. Vaccaro and A.F. Zobaa. Cooperative fuzzy controllers for autonomous voltage regulation in smart grids. *Journal of Ambient Intelligence and Humanized Computing*, 2(1):1–10, 2011.

Y. Wang, Z. Liu, and Q. Xu. New rational parties relying on reputation. *Security and Communication Networks*, 7(7): 1128–1137, 2014.

图 6.12 参考文献列表

表 6.1 \citet 和\citep 的区别

\citet{文献标示}	示 例	\citep{文献标示}	示 例
\citet{jon90}	Jones et al. (1990)	\citep[chap. 2]{jon90}	Jones et al. (1990, chap. 2)
\citep{jon90}	(Jones et al., 1990)	\citep[chap. 2]{jon90}	(Jones et al., 1990, chap. 2)
\citep[see][]{jon90}	(see Jones et al., 1990)	\citep[see][chap. 2]{jon90}	(see Jones et al., 1990, chap. 2)
\citet*{jon90}	Jones, Baker, and Williams (1990)	\citep*{jon90}	(Jones, Baker, and Williams, 1990)

此外还有诸如只引作者名字的\citeauthor 和\citeyear,如表 6.2 所示。

表 6.2 只引用作者和年份

命 令 行	对 应 效 果
\citeauthor{jon90}	Jones et al.
\citeauthor*{jon90}	Jones, Baker, and Williams
\citeyear{jon90}	1990
\citeyearpar{jon90}	(1990)

引用多个参考文献如表 6.3 所示。

表 6.3　引用多个参考文献

命　令　行	对应效果
\citet{jon90,jam91}	Jones et al. (1990); James et al. (1991)
\citep{jon90,jam91}	(Jones et al., 1990; James et al. 1991)
\citep{jon90,jon91}	(Jones et al., 1990, 1991)
\citep{jon90a,jon90b}	(Jones et al., 1990a,b)

不同的 LaTeX 编译环境下有不同的格式,如表 6.4 所示。

表 6.4　编译环境与格式

命　令　行	对应效果
\citet{jon90}	Jones et al. [21]
\citet[chap. 2]{jon90}	Jones et al. [21, chap. 2]
\citep{jon90}	[21]
\citep[chap. 2]{jon90}	[21, chap. 2]
\citep[see][]{jon90}	[see 21]
\citep[see][chap. 2]{jon90}	[see 21, chap. 2]
\citep{jon90a,jon90b}	[21, 32]

此外,还有以上两个命令对应的去掉括号的命令,如表 6.5 所示。

表 6.5　参考文献去掉括号命令

命　令　行	对应效果
\citealt{jon90}	Jones et al. 1990
\citealt*{jon90}	Jones, Baker, and Williams 1990
\citealp{jon90}	Jones et al., 1990
\citealp*{jon90}	Jones, Baker, and Williams, 1990
\citealp{jon90,jam91}	Jones et al., 1990; James et al., 1991
\citealp[pg. 32]{jon90}	Jones et al., 1990, pg. 32
\citetext{priv. comm.}	(priv. comm.)

第 7 章　常见错误与警告

在编译 LaTeX 源文件的过程中，经常会出现一些错误，本章就一些常见的错误，给出错误的原因及解决方案。

7.1　缺少文件错误提示

LaTeX 中最经常出现的错误是有些附件文件没有和源文件 TeX 在同一个文件夹下。解决的方法是，将对应的文件(如 cls、sty 格式的文件)复制到源文件所在的文件夹下即可。一旦出现这样的错误，系统会给出提示，如图 7.1 所示。

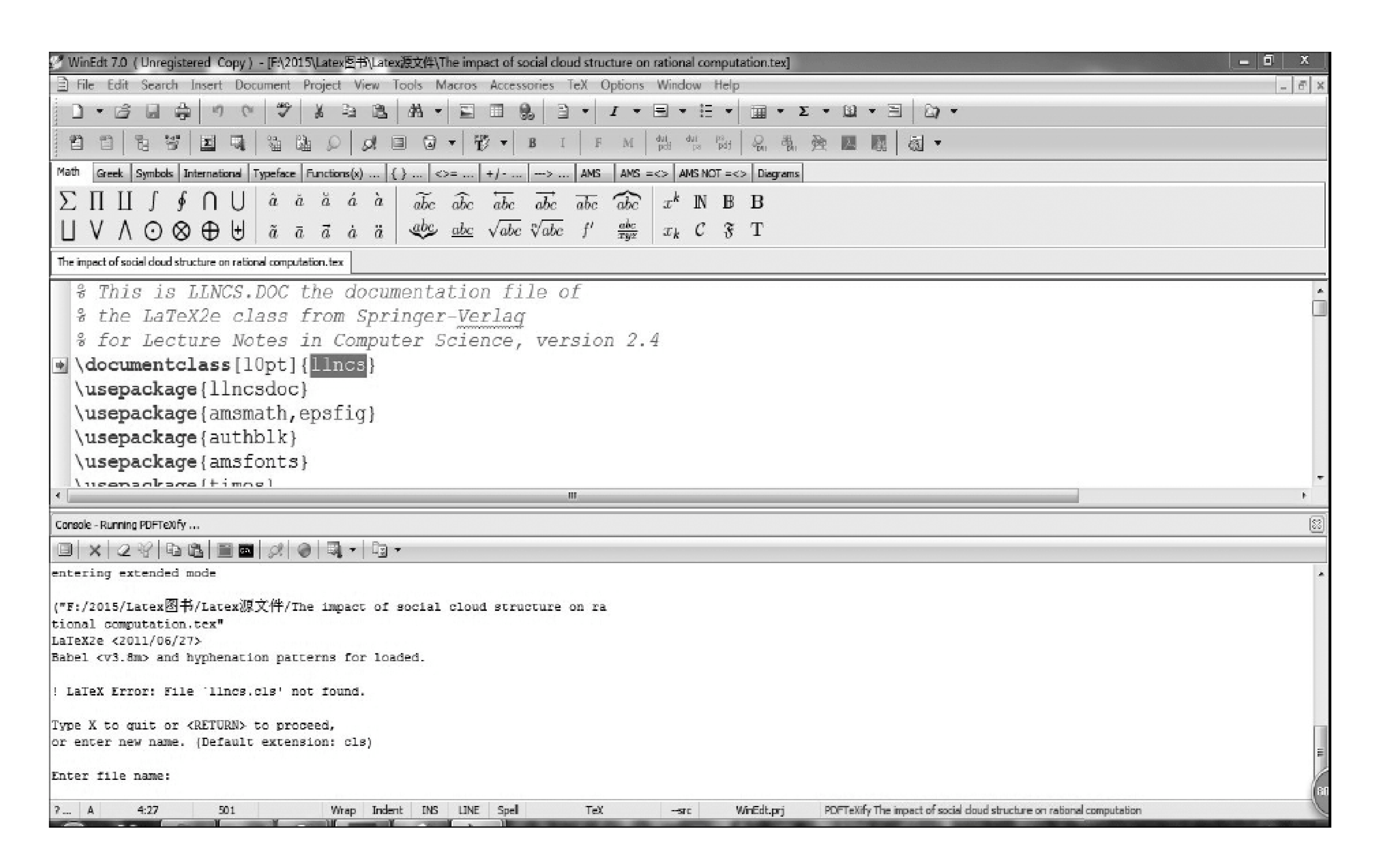


图 7.1　缺少 llncs. cls 文件错误

图中给出的提示是"LaTeX Error: File 'llncs.cls' not found."。另外，有时缺少宏包文件也会出现错误，如图 7.2 所示。图中给出的提示是"LaTeX Error: File 'llncsdoc.sty' not found."。

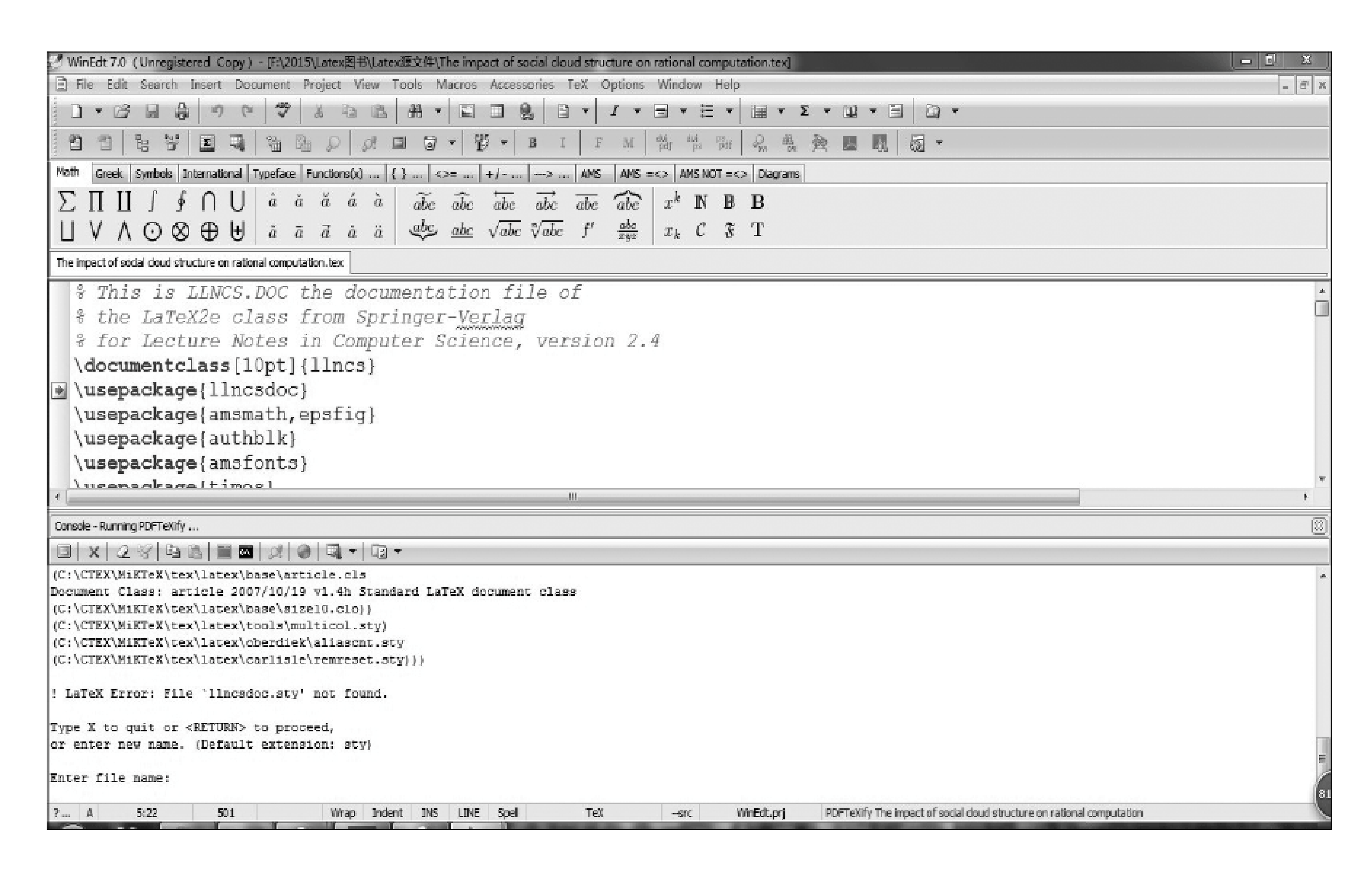


图 7.2　缺少宏包文件错误

除了缺少系统文件以外，有时图形没有与源文件在同一个文件夹中也会出现错误，如图 7.3 所示。图中给出的提示是"LaTeX Error: File 'architecture' not found."。

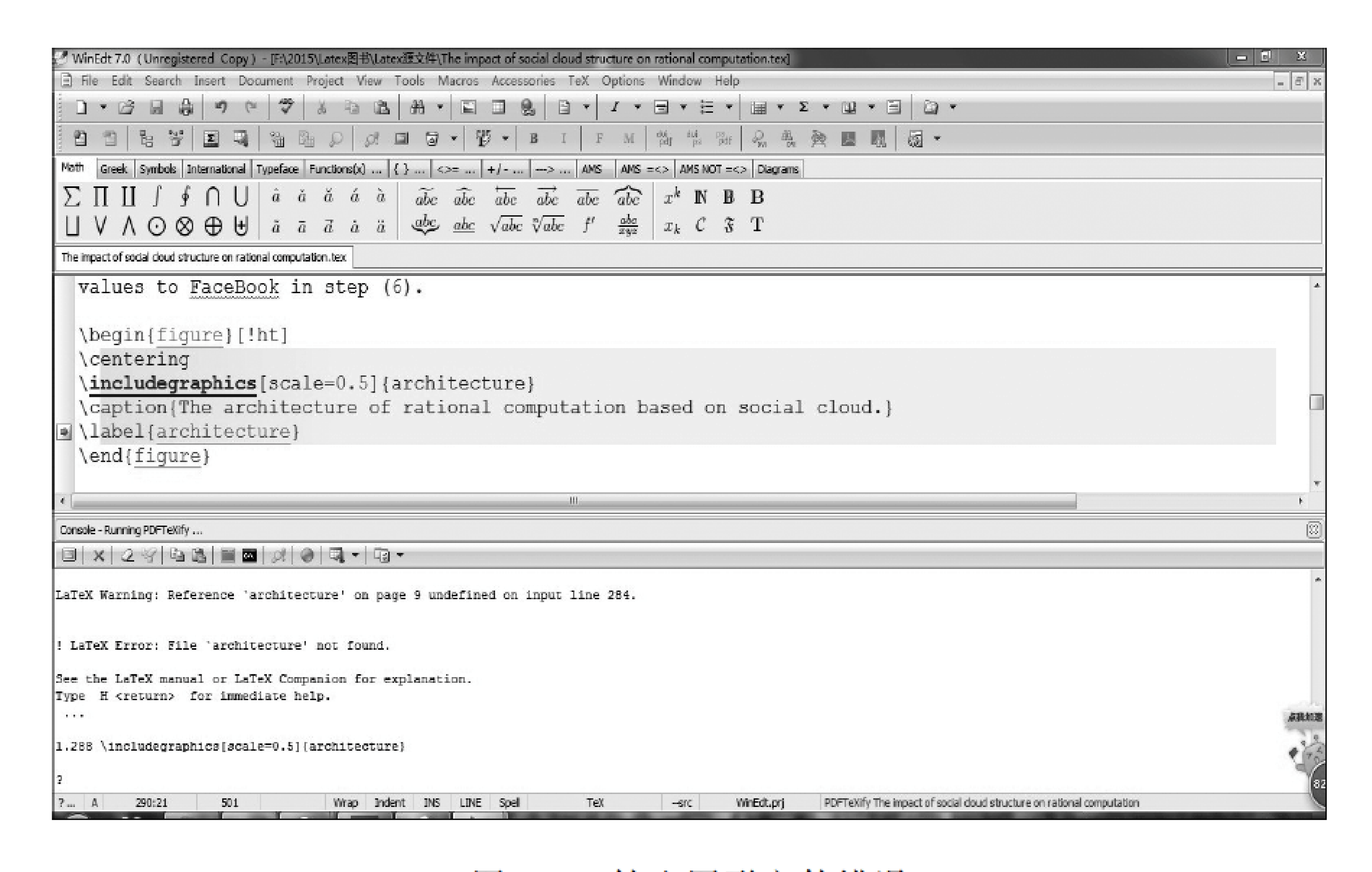


图 7.3　缺少图形文件错误

7.2 数学公式错误提示

LaTeX 的一大优势就在于数学公式排版,但是在排版过程中数学公式也经常出错。最近常出现的错误是 $ 符号没有成对出现,一旦出现这样的错误(见图 7.4),给出的提示是“Missing $ inserted.”,并且还会给出出错的行号,本例中出错的行号是 314 行。

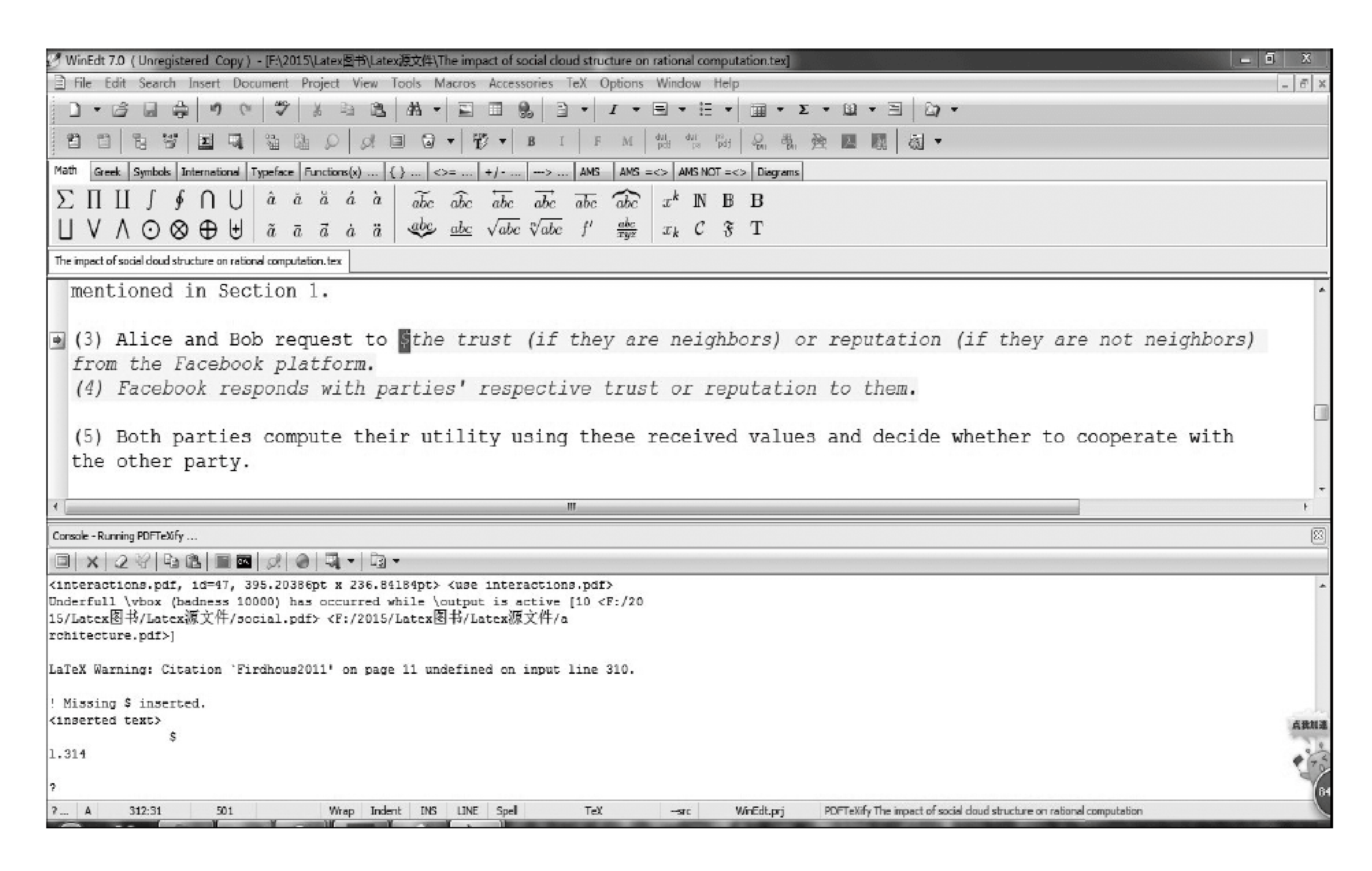


图 7.4　符号 $ 未成对出现错误

另外,有时用到一些特殊的数学符号,但是这些数学符号不是系统默认的,需要宏包支持,如果在正文中引用该符号,但是没有添加相应的宏包,也会出现错误提示。

例如,如果想在正文中显示直立积分号 $\iint_a^b f(x)$,在 Word 中很容易输入,在 LaTeX 中需要输入命令行:$\iint_a^b f(x)$。但是这个数学公式需要用到宏包 amsmath,必须在\begin{document}之前加入宏包\usepackage{amsmath},否则会出现错误,如图 7.5 所示。错误提示是“undefined control sequence”。

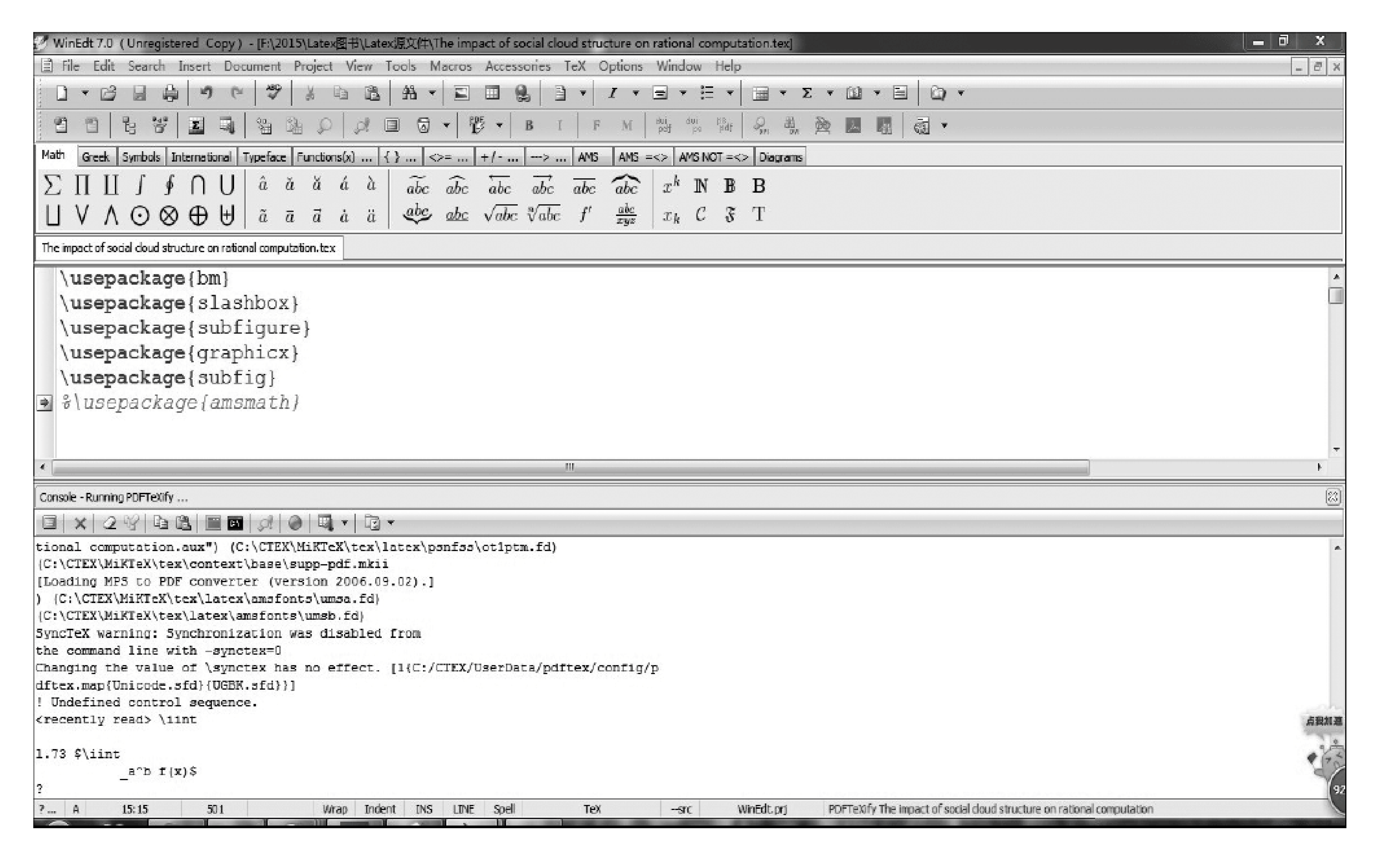


图 7.5　未包含宏包错误

7.3　表格错误提示

一般情况下，模板会给出通用的表格模板，但是给出的模板中的行列数和文章中需要的行列数未必一致，因此需要用户自己更改行和列的数目。在更改的过程中，可能因为表头所规定的列数和用户填入的列数不一致而产生错误。例如，图 4.1 所示的表格是 4 行 4 列的一个表格。其中\begin{tabular}{lccc}表示该表格有 4 列，如果这里用户不小心少输入一个 c，也就是说，规定的是一个 3 列的表格，但是后面的命令行却只指定了 4 列，就会出现错误，如图 7.6 所示。

需要注意的是，如果多输入了列数，但是指定的列数少，是不影响最后编译结果的。

在输入各行信息时，可能会出现漏掉 & 的情况(命令行见图 7.7)，所对应的表格内容就自动左移，如图 7.8 所示。

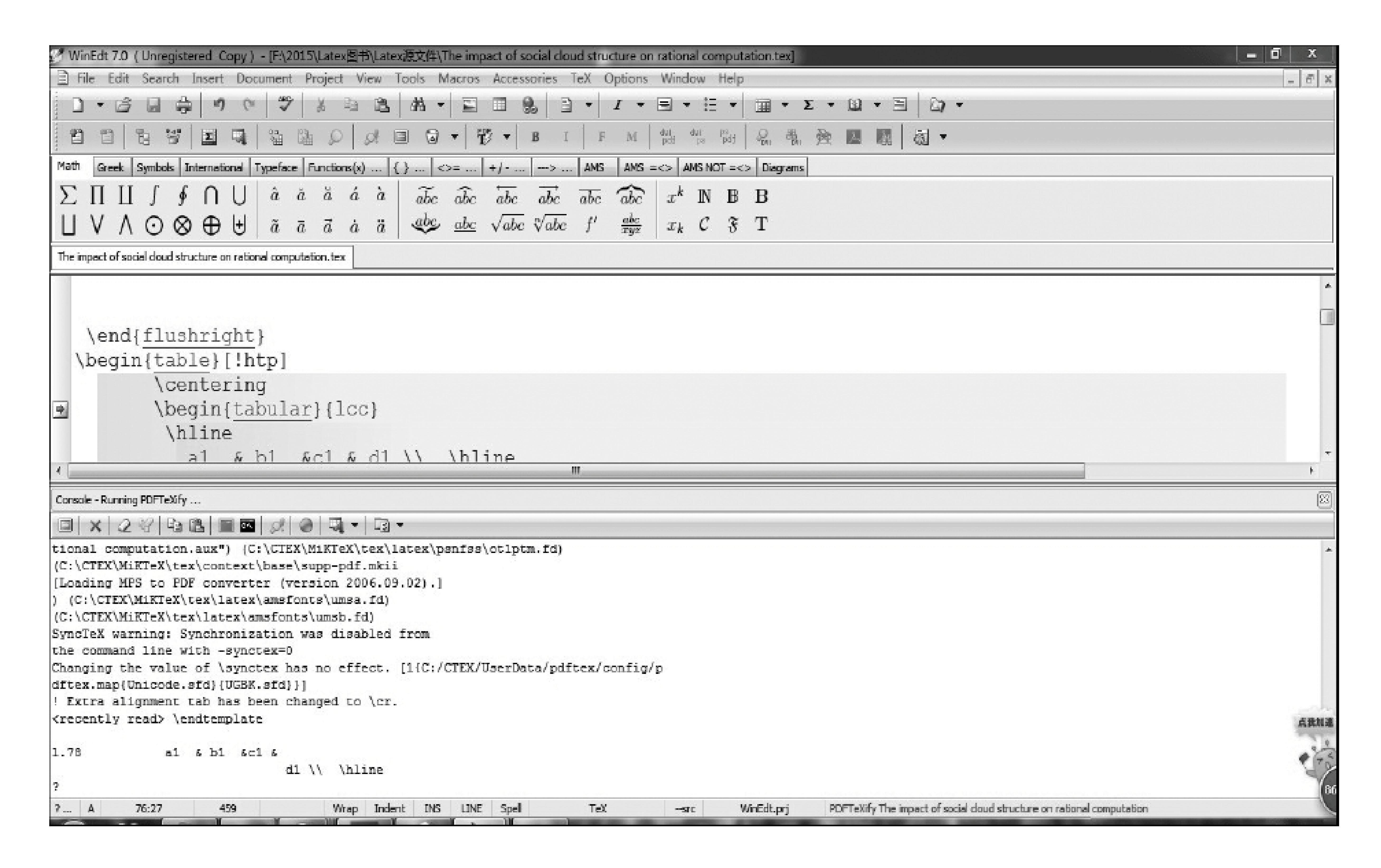


图 7.6　表格列数不一致错误

```
\begin{table}[!htp]
    \centering
    \begin{tabular}{lcccc}
     \hline
       a1  & b1  &c1  & d1 \\  \hline
       a2  & b2  &c2    d2 \\
       a3  & b3  &c3  & d3 \\
       a4  & b4  &c4  & d4 \\   \hline
    \end{tabular}
    \caption{title}
     \label{tab:example}
  \end{table}
```

图 7.7　表格漏掉 & 的命令行示意图

a1	b1	c1	d1
a2	b2	c2	d2
a3	b3	c3	d3
a4	b4	c4	d4

Table 1: title

图 7.8　表格漏掉 & 的效果示意图

7.4　保留字符错误提示

字符 #、$ 、% 、^、&、_ 、{、}、~、\是 LaTeX 的保留字符，它们或者在 LaTeX 中有特定的用处，或者不一定包含在所有的字库中。如果用户直接在文本中使用它们，通常在排版结果中将不会得到这些字符，甚至有时还会提示错误。当然，在这些字符前面加上反

斜线，就可以在文本中得到它们。

除了\#、\$、\%、\^、\{、\}、\&、_、\~之外，另外一些符号可以由特殊的命令或作为重音命令得到。反斜线 \ 不能够通过在其前添加另外的反斜线来得到，相反，\\是一个用来断行的命令。

如果在正文中错误地使用了这些保留字符，就会产生错误。例如本文的例子，作者邮箱是 la_tex2000@126.com。“_”是一个保留字符，不能直接使用，所以会出现错误，如图 7.9 所示。如果想更正这个错误需要在“_”前面加上符号\，再次运行，就不会出现错误了。注意，这里给出的错误提示是“Missing $ inserted”，是因为系统把“_”看作是数学符号。加上符号\后，就把“_”看作正常的下划线符号。

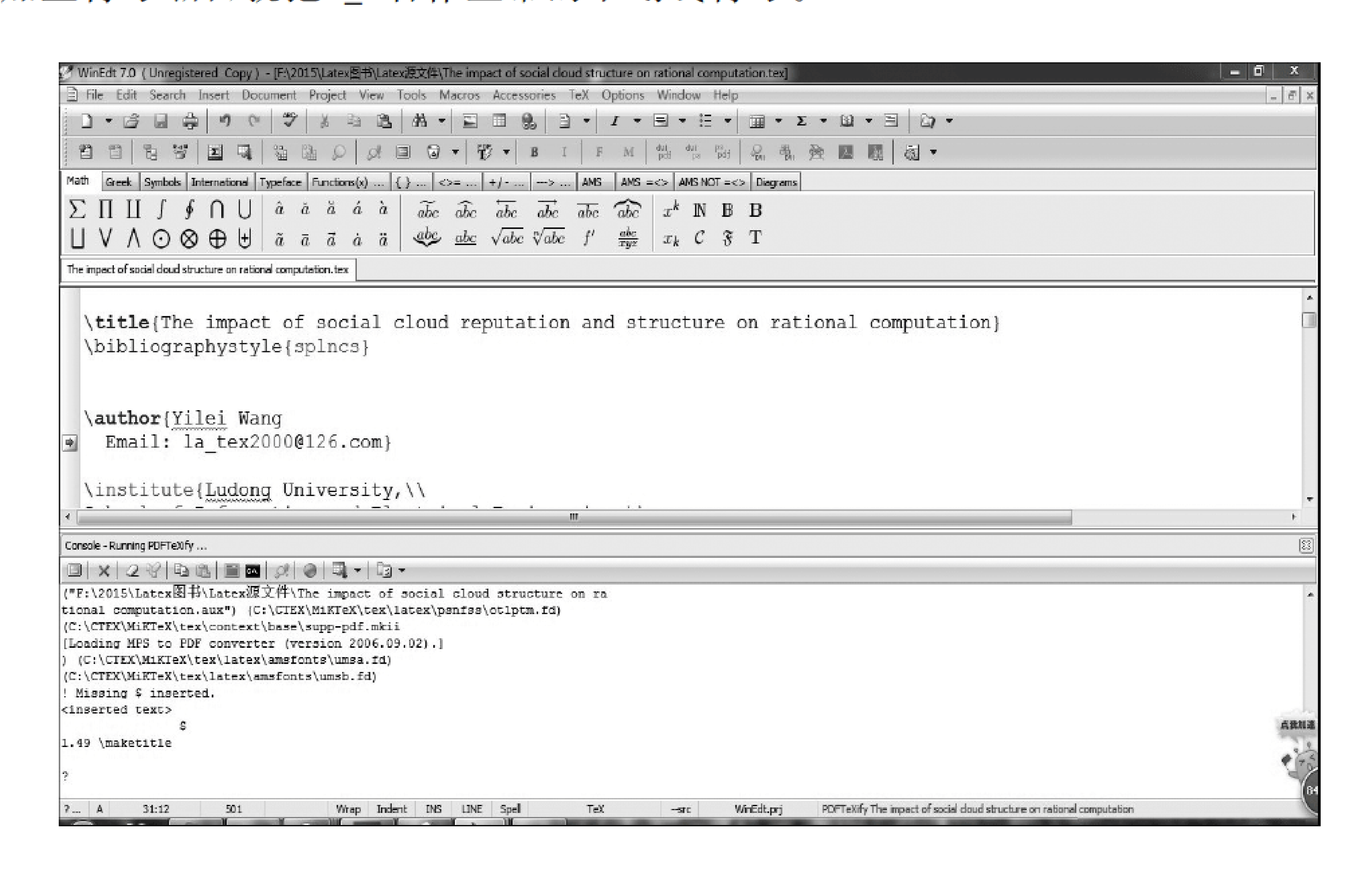


图 7.9 保留字符错误

7.5 其他常见错误提示

(1) 错误类型 e_fileEnd。错误提示信息：File ended while scanning use of \end。错误产生原因：Generally caused because of missing a brace。

(2) 错误类型 e_end。错误提示信息：No message only an asterisk, i. e * 。错误产

生原因：Missing \end{document}。

（3）错误类型 e_illegal。错误提示信息：LaTeX Error：Illegal character in array arg。错误产生原因：Usage of a letter other than r，l and c in tabular environment。

（4）错误类型 e_tab。错误提示信息：！Misplaced alignment tab character &。错误产生原因：Missing \begin{tabular} while using tabular environment。

（5）错误类型 e_backslash。错误提示信息：！Missing \endcsname inserted。错误产生原因：Usage of a backslash in front of the name of an environment，e. g \begin{\itemize}。

（6）错误类型 e_delimiter。错误提示信息：！LaTeX Error：Bad math environment delimiter。错误产生原因：Missing \right immediately after the array environment。

（7）错误类型 e_right。错误提示信息：！Extra \right。错误产生原因：\right has no matching \left OR \end{array} is missing。

（8）错误类型 e_package。错误提示信息：！LaTeX Error：Can only be used in preamble。错误产生原因：Usage of \usepackage outside the preamble。

（9）错误类型 e_math。错误提示信息：！Missing $ inserted。错误产生原因：Missing a starting or ending $ in Math mode，e. g m_e instead of m_e。

（10）错误类型 e_parameter。错误提示信息：！Illegal parameter number in definition of…。错误产生原因：Usage of parameter number greater than the number of parameters defined in \newcommand，e. g \newcommand{\test}[1]{♯3}。

（11）错误类型 e_cmd。错误提示信息：！LaTeX Error：Command … already defined。错误产生原因：Trying to define already existing command，e. g \newcommand{\time}。

（12）错误类型 e_caption。错误提示信息：！LaTeX Error：\caption outside float。错误产生原因：\caption{…} used outside table environment。

（13）错误类型 e_braces。错误提示信息：！Too many }'s。错误产生原因：Missing \begin{table}statement。

（14）错误类型 e_parbox。错误提示信息：！Argument of \@caption has an extra }。

错误产生原因：Usage of \parbox in a \caption。

（15）错误类型 e_item。错误提示信息：! LaTeX Error：Something's wrong—perhaps a missing \item。错误产生原因：Missing \item within enumerate environment。

（16）错误类型 e_fraction。错误提示信息：! Argument of \end has an extra }。错误产生原因：Misuse of fraction cmd e. g \frac{1,2}。

（17）错误类型 e_verb。错误提示信息：! LaTeX Error：\verb ended by end of line。错误产生原因：Newline after \verb，e. g. \verb * dir * 。

（18）错误类型 e_invalid。错误提示信息：! LaTeX Error：Command \end{itemize} invalid in math mode。错误产生原因：Missing $ while using math mode in \itemize。

（19）错误类型 e_equation。错误提示信息：! Display math should end with $ $ 。错误产生原因：Usage of $ $ inside equation mode。

（20）错误类型 e_column。错误提示信息：! Misplaced \omit。错误产生原因：Usage of \newcommand and \multicolumn within tabular environment。

（21）错误类型 e_subscript。错误提示信息：! Double subscrip。错误产生原因：Usage of double subscript。

（22）错误类型 e_cls。错误提示信息：! LaTeX Error：File 'artcle. cls' not found。错误产生原因：Missing . sty or . cls file。

（23）错误类型 e_nofile。错误提示信息：! LaTeX Error：File 'file1. tex' not found。错误产生原因：Missing file1. tex，e. g. \input{file1. tex}。

（24）错误类型 e_sty。错误提示信息：! LaTeX Error：File 'anysize1. sty' not found。错误产生原因：Use of unavailable package。

（25）错误类型 e_doc_class。错误提示信息：! LaTeX Error：Can be used only in preamble。错误产生原因：Usage of \documentclass outside preamble。

（26）错误类型 e_circle。错误提示信息：! LaTeX Error：Command \circle invalid in math mode。错误产生原因：Usage of \circle in math mode。

（27）错误类型 e_picture。错误提示信息：! Use of \pictur@ doesn't match its definition。错误产生原因：Bad parameter to \picture。

(28) 错误类型 e_line。错误提示信息：! Use of \put dosen't match its definition。错误产生原因：Badly formatted \line directive。

(29) 错误类型 e_line_arg。错误提示信息：! LaTeX Error：Bad \line or \vector argument。错误产生原因：Bad \line parameter。

(30) 错误类型 e_counter。错误提示信息：! LaTeX Error：No counter '10' defined。错误产生原因：Counter undefined。

(31) 错误类型 e_outer。错误提示信息：! LaTeX Error：Not in outer par mode。错误产生原因：Using figure inside parbox。

(32) 错误类型 e_minipage。错误提示信息：! LaTeX Error：Not in outer par mode。错误产生原因：Using figure minipage。

(33) 错误类型 e_lost。错误提示信息：! LaTeX Error：Float(s) lost。错误产生原因：Counter undefined。

(34) 错误类型 e_lonely。错误提示信息：! LaTeX Error：Lonely \item—perhaps a missing list environment。错误产生原因：Usage of \item outside list environment。

(35) 错误类型 e_parg。错误提示信息：! LaTeX Error：Missing p-arg in array arg。错误产生原因：Missing p argument in tabular environment。

(36) 错误类型 e_hash。错误提示信息：! You can't use 'macro parameter character ＃' in vertical mode。错误产生原因：Usage of ＃ in normal mode。

(37) 错误类型 e_enlarge。错误提示信息：! LaTeX Error：Suggested extra height (14454.0pt) dangerously large。错误产生原因：Too big a number given in \enlargethispage。

(38) 错误类型 e_deftab。错误提示信息：! LaTeX Error：Undefined tab position。错误产生原因：Undefined tabbing。

(39) 错误类型 e_pushtab。错误提示信息：! LaTeX Error：\pushtabs and \poptabs don't match。错误产生原因：Unequal numbers of push and pop tabs。

(40) 错误类型 e_overtab。错误提示信息：! LaTeX Error：Tab overflow。错误产生原因：Too many \=in tabbing environment。

(41) 错误类型 e_nest。错误提示信息：! LaTeX Error：Too deeply nested。错误产生原因：Too many list environments。

(42) 错误类型 e_eqnarray。错误提示信息：! LaTeX Error：Too many columns in eqnarray environment。错误产生原因：More than three columns in eqnarray。

(43) 错误类型 e_classpkg。错误提示信息：! LaTeX Error：\usepackage before \documentclass。错误产生原因：Usage of usepackage before loading documentclass。

(44) 错误类型 e_load。错误提示信息：! LaTeX Error：Two \LoadClass commands。错误产生原因：More than one load class command。

(45) 错误类型 e_require。错误提示信息：! LaTeX Error：\RequirePackage or \LoadClass in Options Section。错误产生原因：RequirePackage may not be used with \DeclareOption。

(46) 错误类型 e_twoclass。错误提示信息：! LaTeX Error：Two \documentclass or \documentstyle commands。错误产生原因：More than one documentclass declaration。

(47) 错误类型 e_font。错误提示信息：! LaTeX Error：This NFSS system isn't set up properly。错误产生原因：Invalid font used in \DeclareErrorFont。

(48) 错误类型 e_superscript。错误提示信息：! Double superscript。错误产生原因：Usage of two superscripts for the same variable，e.g. 2^3^4。

(49) 错误类型 e_clash_opt。错误提示信息：! LaTeX Error：Option clash for package csvtools。错误产生原因：Clashing options for the same package。

(50) 错误类型 e_unknown_opt。错误提示信息：! LaTeX Error：Unknown option…for package…。错误产生原因：Unkown option for a package。

(51) 错误类型 e_hyphenation。错误提示信息：! Improper \hyphenation will be flushed。错误产生原因：Improper parameter to \hyphenation。

(52) 错误类型 e_stack_size。错误提示信息：! TeX capacity exceeded，sorry [main memory size = 1000000]。错误产生原因：Overflow of buffer due to mistake in command definition。

(53) 错误类型 e_environment。错误提示信息：! LaTeX Error：Environment …

undefined。错误产生原因：Undefined environment。

(54) 错误类型 e_midline。错误提示信息：! LaTeX Error：\< in mid line。错误产生原因：Command \< may appear only at the beginning of a line。

(55) 错误类型 e_infinite。错误提示信息：Goes into infinite loop。错误产生原因：Usage of \\strut\hrule。

(56) 错误类型 e_des。错误提示信息：! LaTeX Error：There's no line here to end。错误产生原因：Usage of \\ at the end of a long label in 'description' environment。

(57) 错误类型 e_center。错误提示信息：! LaTeX Error：There's no line here to end。错误产生原因：Usage of \\ after the heading line in 'center' environment。

(58) 错误类型 e_foot。错误提示信息：! Argument of \@sect has an extra }。错误产生原因：Usage of a fragile command 'footnote' within \section。

(59) 错误类型 e_ragged。错误提示信息：! Argument of \@caption has an extra }。错误产生原因：Usage of \\ within \raggedright or \raggedleft environment。

(60) 错误类型 e_and。错误提示信息：! Extra alignment tab has been changed to \cr。错误产生原因：Too many &s in a row of a table or array or eqnarray。

(61) 错误类型 e_cline。错误提示信息：! Extra alignment tab has been changed to \cr。错误产生原因：Reference no non existing column in \cline。

(62) 错误类型 e_col。错误提示信息：! Extra alignment tab has been changed to \cr。错误产生原因：Usage @ in tabular * environment。

(63) 错误类型 e_num。错误提示信息：! Missing number treated as zero。错误产生原因：Usage of non numeric parameter after \\。

(64) 错误类型 e_asterisk。错误提示信息：Missing * at the end of the line。错误产生原因：* is not printed when used without brace after \\。

(65) 错误类型 e_pbox_miss。错误提示信息：! Missing number，treated as zero。错误产生原因：\parbox[t]{}. Missing argument to parbox。

(66) 错误类型 e_mis_circle。错误提示信息：! Missing number，treated as zero。错误产生原因：Missing numeric parameter to \circle。

(67) 错误类型 e_list。错误提示信息：! Argument of \lst@next has an extra }。错误产生原因：Usage of 1stlisting inside fragile command \parbox。

(68) 错误类型 e_capacity。错误提示信息：! TeX capacity exceeded，sorry [input stack size=1500]。错误产生原因：Usage of 1stlisting inside fragile command \parbox。

(69) 错误类型 e_runaway。错误提示信息："Runaway argument?"。错误产生原因：Generally because of missing braces，e. g \cline{1-2 instead of \cline{1-2}。

(70) 错误类型 e_verbatim。错误提示信息："Runaway argument?"。错误产生原因：Usage of verbatim within scope of another command e. g：\ifthenelse。

(71) 错误类型 e_undefined。错误提示信息：! Undefined control sequence。错误产生原因：Usage of an unknown command。

(72) 错误类型 e_footnote。错误提示信息：! Undefined control sequence。错误产生原因：Usage of \footnote within \footnote。

(73) 错误类型 e_integral。错误提示信息：! Missing { inserted。错误产生原因：Integral bounds are malformed。

(74) 错误类型 e_zeta。错误提示信息：! Missing { inserted。错误产生原因：Extra subscript before integral upper limit term。

(75) 错误类型 e_bezier。错误提示信息：! Illegal unit of measure (pt inserted)。错误产生原因：Missing numeric argument to \qbezier。

(76) 错误类型 e_too_bezier。错误提示信息：! Illegal unit of measure (pt inserted)。错误产生原因：Too many arguments to \qbezier。

(77) 错误类型 e_unit。错误提示信息：! Illegal unit of measure (pt inserted)。错误产生原因：\parbox[t]{2}. Illegal unit of second parameter。

(78) 错误类型 e_symfoot。错误提示信息：! LaTeX Error：Counter too large。错误产生原因：More than 9 footnotes when using symbolic footnotes。

(79) 错误类型 e_large_count。错误提示信息：! LaTeX Error：Counter too large。错误产生原因：Trying to display a corresponding letter for a counter vallue >26。

（80）错误类型 e_begin。错误提示信息：！LaTeX Error：Missing \begin{document}。错误产生原因：Either text has been placed before \begin{document} or \begin{document} is missing。

（81）错误类型 e_margin。错误提示信息：！LaTeX Error：Missing \begin{document}。错误产生原因：Misuse of \marginsize。

参 考 文 献

[1] 桑大勇，王瑛．科技文献排版系统 LaTeX 入门与提高[M]．武汉：武汉大学出版社，2001．

[2] 胡伟．LaTeX2e 完全学习手册[M]．北京：清华大学出版社，2011．

[3] 刘海洋．LaTeX 入门[M]．北京，电子工业出版社，2013．

[4] 刘小平．论文排版实用教程——Word 与 LaTeX[M]．北京：清华大学出版社，2015．

[5] 陈志杰，等．LaTeX 入门与提高[M]．北京：高等教育出版社，2006．

[6] Herbert Voss. Typesetting Tables with LaTeX[M]. UIT Cambridge Ltd.，2010.

[7] 罗振东，葛向阳．排版软件 LaTeX 简明手册[M]．2 版．北京：电子工业出版社，2004．

[8] 科普卡，等．LaTeX 实用教程[M]．4 版．北京：机械工业出版社，2005．

[9] 李平．LaTeX2 及常用宏包使用指南[M]．北京：清华大学出版社，2004．

[10] ICON HealthPublications. Latex Allergy—A Medical Dictionary，Bibliography [M]. Icon GroupInternational，1988.

[11] Michel Goossens. The LaTeX Graphics Companion[M] 2nd Edition. Addison-Wesley Professional，2007.

[12] Apostolos Syropoulos. Digital Typography Using LaTeX[M]. Springer，2002 .